

Ângelo Boaventura Vaz Semedo

Utilização de Plataformas Open Source

Caso de estudo Empresas de Base Tecnológica e
Instituições de Ensino Superior da Cidade da Praia

Universidade Jean Piaget de Cabo Verde

Campus Universitário da Cidade da Praia
Caixa Postal 775, Palmarejo Grande
Cidade da Praia, Santiago
Cabo Verde

12.6.14

“O software proprietário é dependência e isso leva à colonização eletrônica. As empresas do software proprietário querem colonizar todos os países: eles tomaram os Estados Unidos, Europa e outros lugares do mundo”

(Richard Stallman)

Ângelo Boaventura Vaz Semedo

Utilização de Plataformas Open Source

Caso de estudo Empresas de Base Tecnológica e
Instituições de Ensino Superior da Cidade da Praia

Ângelo Semedo, autor da monografia intitulada *Utilização de plataformas Open Source*, declaro que, salvo fontes devidamente citadas e referidas, o presente documento é fruto do meu trabalho pessoal, individual e original.

Cidade da Praia aos 15 de Maio de 2013
Ângelo Boaventura Vaz Semedo

Memória Monográfica apresentada à Universidade Jean Piaget de Cabo Verde como parte dos requisitos para a obtenção do grau de Licenciatura em Engenharia de sistemas e informática.

Sumário

Este é o meu trabalho do fim de curso de Engenharia de Sistemas e Informática, no grau de licenciatura que tem como tema “*Utilização de Plataformas Open Source*” onde vou abordar a utilização de plataformas *Open Sources* na área de TI mas concretamente no desenvolvimento do software em geral o que engloba todas as fases de desenvolvimento do software.

De um modo geral esta memória visa fazer uma pequena publicitação dos softwares *Open Sources*, sabendo que é muito comum encontrarmos grandes softwares *Open Source* mais pouco utilizados em relação a um outro software do mesmo tipo mas que é proprietário, como no caso de Windows e Linux.

Pretende-se analisar as principais características de softwares *Open Source* que dificultam a sua utilização por parte dos profissionais de empresas de base tecnológica e chefe de laboratórios informático das instituições de ensino superior da Cidade da Praia, também pretende saber o porque da dominação dos softwares proprietários em países em desenvolvimento em geral.

Entretanto vamos apresentar as principais melhorias que empresas de desenvolvimento de softwares *Open Source* devem fazer para aumentar a utilização dos seus softwares em comparação com o software proprietário.

Palavras-chaves: *Open Source*, Licença de softwares *Open Source*, Software Proprietário, *Closed Source*, Tecnologias *Open Source*;

Dedicatória

Dedico esta memória ao meu irmão (Jorge Semedo) e a minha mãe (Júlia Andrade).

Agradecimentos

Em primeiro lugar agradeço a Deus pelo dom da vida e por todos os conhecimentos adquirido ao longo desta caminhada.

Agradeço ao meu professor e orientador, Mestre Stefan Manuel Monteiro, por todo o apoio e paciência durante a realização desse trabalho.

Agradeço aos meus professores, o Dr. Isaías Barreto da Rosa, Engenheiro Mário Monteiro, por apoio e paciência durante a realização desse trabalho e pelo material fornecido.

Gostaria de agradecer toda a minha família que me deu força desde o início da minha formação.

Aos meus colegas, que me apoiaram quando necessário com documentos, links, opiniões no momento em que mais precisei etc. obrigado.

À Universidade Jean Piaget de Cabo Verde por me ter proporcionado as condições desejáveis para a realização dos meus estudos.

Enfim, agradeço a todos que de uma forma ou outra colaboraram para que hoje eu pudesse apresentar esta memória monográfica, em especial á Rosana Moniz.

Conteúdo

Introdução.....	16
1.1 Objetivos.....	17
1.2 Motivação	18
1.3 Metodologias	18
1.4 Pergunta de partida	19
1.5 Hipótese.....	19
1.6 Estrutura do trabalho	19
Capítulo 1: Software Open Source e software proprietário.....	21
1 Open Source	21
1.1 Definição	22
1.2 Historial	24
1.3 Open Source e Free software (Diferença)	27
2 Software proprietário.....	28
2.1 Definição	28
2.2 Historial	30
3 Licença Open Source.....	31
3.1 Problemas relacionados com as licenças Open Sources.....	39
4 Open Source versus Closed Source	40
4.1 Custo.....	41
4.2 Serviço e suporte	42
4.3 Inovação	43
4.4 Usabilidade	44
4.5 Segurança	45
4.6 Desenvolvimento	47
4.7 Vantagens	48
4.8 Desvantagens.....	49
5 Software Open Source nos países em desenvolvimento como Cabo Verde	50
5.1 Oportunidade para países em desenvolvimento	52
6 Considerações finais.....	53
Capítulo 2: Tecnologias Open Source.....	54
1 Web browsers	55
1.1 Mozilla Firefox	55
2 IDEs.....	57
2.1 Eclipse	57
2.2 Netbeans	58
3 Servidor Web.....	59
3.1 Apache	60
4 Editores.....	60
4.1 EMACS	61
4.2 GIMP	62
5 Base de dados	64
5.1 Mysql.....	64
5.2 PostgreSQL	65
6 Sistemas operativos	66
6.1 Linux.....	67

6.2	História do Linux.....	68
6.3	Distribuições Linux	70
7	Considerações finais	71
Capítulo 3: Estudo de Caso		73
1	Introdução.....	73
1.1	Instituições do ensino superior que participaram	74
1.2	Empresas base tecnológica que participaram	75
2	Softwares Open Sources em Cabo Verde.....	76
2.1	Nas empresas de base tecnológica.....	78
2.2	Nas instituições de ensino superior	85
3	Outras informações obtidas	89
Conclusão		94
A	Anexo	102
A.1	Questionário às Empresas de base tecnológica	102
A.2	Questionário às Universidades	106

Tabelas

Tabela 1: Comparação entre filosofia OS e SP, adaptado de Engelfriet (2003).	43
Tabela 2: Principais práticas de usabilidade utilizadas no desenvolvimento de software Open Source (Santos, 2012).	44
Tabela 3: Número de vulnerabilidade de SOS e SP adaptado de Schryen (2009).	46
Tabela 4: Comparação entre os modelos Catedral e Bazar (Farias, 2011).	48
Tabela 5: Ranking dos principais SGBD de agosto de 2013 (Ranking-DB-Engines, 2013). ..	65
Tabela 6: Vantagens do Open Source sobre o Closed Source segundo os inquiridos.	92
Tabela 7: Desvantagens do Open Source em relação ao Closed Source segundo os inquiridos.	92

Figuras

Figura 1: Símbolo do copyright (Wikimedia, 2013b).	33
Figura 2: Símbolo de copyleft. (Wikimedia, 2013a)	33
Figura 3: Licenças Open Source, projetos e autores, adaptado de Meeker (2008)	34
Figura 4: Parcela de mercado dos principais servidores Web (Netcraft, 2013)	35
Figura 5: Índice de pesquisa de custos e suporte dos softwares Open Sources (Mandelbaum, 2012)	42
Figura 6: A utilização do sistema operativo de desktop (I. B. da Rosa & Mendes, 2010)	51
Figura 7: Razões da utilização de sistemas operativo de desktop (I. B. da Rosa & Mendes, 2010)	51
Figura 8: Top 5 browser de 2008 à 2013 (Stats, 2013a)	56
Figura 9: Interface do Eclipse (ApacheORG, 2013)	58
Figura 10: Interface do Netbeans 6.0 (NetBeansORG, 2011)	59
Figura 11: Interface do EMAC 23.1.1 (GNU, 2013)	61
Figura 12: Interface do GIMP 2.8.10 (fonte: http://www.gimp.org/screenshots/)	62
Figura 13: Tux, mascote do Linux (Wikimedia, 2013c).	67
Figura 14: Top 7 sistemas operativos 2008 à 2013 (Stats, 2013b)	68
Figura 15: Índice de utilização de sistemas operativo em Cabo Verde (Stats, 2013b)	77
Figura 16: Percentagem da utilização do servidor apache nas empresas de base tecnológica na Cidade da Praia	78
Figura 17: Servidor Apache vs. IIS Windows nas empresas de base tecnológica na Cidade da Praia	79
Figura 18: Índice de utilização do Mozilla Firefox nas empresas de base tecnológica na Cidade da Praia	79
Figura 19: Google Chrome vs. Mozilla nas empresas de base tecnológica na Cidade da Praia	80
Figura 20: Utilização do SO Windows pelas empresas de base tecnológica na Cidade da Praia	81
Figura 21: Utilização do Linux pelas empresas de base tecnológica na Cidade da Praia.	81
Figura 22: Critérios de avaliação de sistemas operativos nas empresas de base tecnológica na Cidade da Praia.	82
Figura 23: Índice de utilizadores que pagam pela licença do Windows nas empresas de base tecnológica na Cidade da Praia.	83
Figura 24: Índice de utilização do GIMP nas empresas de base tecnológica na Cidade da Praia.	84
Figura 25: Photoshop vs. GIMP.	84
Figura 26: Utilização de servidor apache por parte das universidades na Cidade da Praia.	86
Figura 27: Top 5 browser mais utilizados em Cabo Verde de 2010 à 2013.	87
Figura 28: Google Chrome vs. Mozilla Firefox nas universidades da Cidade da Praia.	87
Figura 29: Índice de utilização do Windows por parte das instituições na Cidade da Praia.	88
Figura 30: Índice de utilização do GIMP pelas instituições na Cidade da Praia.	89

Abreviaturas

API – Application programming interface
ASF – Apache Software Foundation
BPM – Bitmap Image File
BSD – Berkeley Software Distribution
CD – Compact Disc
CMS – Sistema de Gerenciamento de Conteúdo
DCU – Designer Centrado no Utilizador
FSF – Free Software Foundation
GIF – Graphics Interchange Format
GIS – Geographic Information System
GNU – GNU is Not Unix
GPL – General Public License
HTML – Hyper Text Markup Language
IBM – International Business Machines
IOS – iPhone Operating System
JPEG - Joint Photographic Experts Group
JSP – Java Server Pages
LINUX – Linux Is Not Unix
MPEG - Moving Picture Experts Group
NOSI – Núcleo Operacional para a Sociedade de Informação
OpenDS – Open Source directory service
OSI – Open Source Initiative
PDF – Portable Document Format
PHP – Hypertext Preprocessor
PNG – Portable Network Graphics
POSIX – Portable Operating System Interface
SGBD – Sistema de Gerenciamento de Banco de Dados
SGBD – Sistema Gerenciador de Banco de Dados
SGBDR – Sistema Gerenciador de Banco de Dados Relacional
SOS – Software Open Source
SP – Software Proprietário

SWT – Standard Widget Toolkit

TI – Tecnologias de Informação

TIFF – Tagged Image File Format

UC – University of California

UNIX – Uniplexed Information and Computing System

XML – Extensible Markup Language

XPM – X Pixmap

Glossário

AbiWord – Processador de texto.

Apache – Servidor *Web*.

Audacity – Editor de áudio.

Blender – Programa para modelação 3D/ aplicação de animação.

Celtix – É um processador de textos especialmente para roteiros.

Dia – É um aplicativo para criação de diagramas.

Distro – Qualquer sistema operativo que utiliza o Kernel Linux, pode ser chamado também de distribuição Linux.

Drupal – É um framework modular e um sistema de gerenciamento de conteúdo (CMS) escrito em PHP.

Eclipse – É um IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de *plugins* como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android.

EMACS – Editor de texto.

Firefox – É um navegador *web (browser)*.

GanttProject – Programa para gestão de projeto.

GIMP – Programa para editar imagens.

Gnumeric – É um aplicativo utilizado para criar *planilhas* eletrônicas e gerar gráficos.

Hardware – São periférico de um computador ou seja a parte física de um computador, com unidade central de processamento, à memória e aos dispositivos de entrada e saída.

ImageMagic – Editor de imagens.

Inkscape – Ilustração / aplicação de desenho.

Kernel – É o componente central do sistema operativo da maioria dos computadores; ele serve de ponte entre aplicativos e o processamento real de dado feito a nível de hardware.

Koffice – É uma versão de suíte de escritório, distribuída junto com a interface gráfica KDE para Linux.

Komposer – *Web* editor.

Linux – Sistema operativo / Kernel.

Moodle – Administração de conteúdo.

NeoOffice - Processador de texto, apresentação e aplicação de folha de cálculo.

Netbeans – É um ambiente de desenvolvimento integrado para desenvolvedores de software nas linguagens Java, C, C++, PHP, Groovy, Ruby, entre outras.

Open Office – Processador de texto, apresentação e aplicação de folha de cálculo.

Open Project – Gestor de projetos.

Plugin – É um programa de computador utilizado para adicionar funções a outros programas maiores, provendo alguma funcionalidade especial ou muito específica.

Rhythmbox – Leitor de música, gestor de iPod.

Scribus – Auto edição de publicação.

Tellico – É um aplicativo do KDE para a organização de várias coleções. Ele fornece modelos padrão para livros, bibliografias, vídeos, música, jogos de vídeo, moedas, selos, cartões comerciais, revistas em quadrinhos, e vinhos.

Thunderbird – Leitor de correios eletrónico,

Introdução

Vivemos numa era em que a informática está presente em quase tudo e pessoas que não estão familiarizadas com novas tecnologias sentem-se um pouco desintegradas sabendo que o mundo nesse momento é movido a base de computadores (Informática), e a informática leva nos a falar de softwares, visto que o software é uma peça muito importante na informática sabendo que no caso dos computadores só é funcional se encontra alguns software nele instalado.

Computadores precisam constantemente de softwares o que nem sempre não apresentam um preço amigável, e também pode não satisfazer todas as nossas necessidades, é aí que entra o softwares *Open Sources*, que segundo Fuggetta (2003) tem uma licença que nos possibilita fazer alterações de acordo com as nossas necessidades.

Por falar do *Open Source* surge muitas questões como “softwares *Open Sources* são gratuitos?”, “Quais são os principais softwares *Open Source*?”, “Os softwares *Open Sources* possuem licença?”, essas são algumas questões que pretendem responder no decorrer do trabalho.

O presente trabalho visa apresentar o estado atual das ferramentas *Open Source* para ambientes de desenvolvimento de software e nas instituições de ensino superior e nas empresas de base tecnológica da Cidade da Praia, analisando as características e funcionalidades e o índice de utilização de cada uma delas.

Este trabalho visa ajudar os leitores a ter uma visão mais abrangente sobre os softwares *Open Sources*, apresentando os tipos de licença que estão por detrás das principais tecnologias

Open Source, apresentando os principais vantagens e desvantagens dos softwares *Open Source* em comparação com os softwares proprietários, e também apresentado as principais oportunidades que os softwares *Open Source* trás para países em desenvolvimento com no caso de Cabo Verde. O Weerawarana et al. (2004) afirma que nos países em desenvolvimento os softwares proprietários e que dominam o mercado, visto que os softwares *Open Source* não possuem recursos suficiente para fazer uma forte campanha de marketing em comparação com os softwares proprietário como no caso de Windows, e este trabalho vai mostrar que existem grande software *Open Source* que podem ser utilizado a nível profissional. De acordo com Sabino e Kon (2009) uma das principais vantagens dos softwares *Open Source* é o baixo custo mas como vê-se mais acima nos países em desenvolvimento como Cabo Verde parece que não aproveitam dessa vantagem, e nesse trabalho vamos procurar saber o real motivo de que softwares proprietário dominaram o mercado nos países em desenvolvimento sabendo que estes possuem menos recursos em comparação com países desenvolvidos.

1.1 Objetivos

1.1.1 *Objetivo Geral*

Analisar as peculiaridades dos softwares *Open Sources*, suas adequações aos países em via desenvolvimento como Cabo Verde, bem como o estado atual da utilização desses tipos de softwares nas instituições de ensino superior e de empresas base tecnológica da Cidade da Praia e os ganhos que trazem.

1.1.2 *Objetivos Específicos*

- Identificar as principais características das plataformas *Open Source* que dificultam a sua utilização;
- Verificar e apresentar as especificidades dos softwares *Open Source* versus software proprietário;
- Apresentar as especificidades da utilização dos softwares *Open Sources*;

- Estudar o estado atual da utilização dos softwares *Open Source* nas instituições do ensino superior e de empresas de base tecnológica da Cidade da Praia;
- Apresentar as eventuais mais-valias do software *Open Source* para países como Cabo Verde;
- Estudar e apresentar quais as tecnologias *Open Source* mais utilizadas pelas instituições de ensino superior na Praia e quais os motivos.

1.2 Motivação

A motivação para a realização deste trabalho prende-se com o facto de Cabo Verde ser um país em desenvolvimento médio. Com poucos recursos financeiro e tecnológico, mas engajado no seu desenvolvimento tecnológico. Com isso precisa abraçar um paradigma que permite a sustentabilidade desse desenvolvimento, ou melhor abraçar o movimento *Open Source*.

Um outro fator de motivação na escolha do tema é o de adquirir mais conhecimento sobre os softwares *Open Sources* uma vez que este tem sido o foco de muitas discussões e debate. Curiosidade de conhecer principais oportunidades (vantagens) que um software *Open Source* oferece a países em desenvolvimento como Cabo Verde.

1.3 Metodologias

Para realização deste trabalho inicialmente fizemos pesquisas coletando bibliografias de grande interesse que serviram com pilar na realização do trabalho. Utilizamos o Mendeley um software para gestão de bibliografia.

Na realização do caso prático utilizou-se as seguintes técnicas:

- Escolha da amostra, num universo de 120 empresas de base tecnológica na Cidade da praia escolhemos 15 aleatoriamente que é cerca de 8% do total, nas instituições de ensino num total de 5 escolhemos 3 que é cerca de 60% do total;
- Construímos dois questionários, um para ser aplicado nas empresas de desenvolvimento de software e o outro para ser aplicado nas instituições de ensino superior na Cidade da Praia;

- De seguida procedeu-se a interpretação dos resultados onde utilizamos o SPSS.

1.4 Pergunta de partida

Porquê que os softwares proprietários são mais utilizados que os softwares *Open Source*, sabendo que os softwares proprietários são mais dispendioso que os softwares *Open Source*, em um país como Cabo Verde?

1.5 Hipótese

Isto deve-se ao facto de em Cabo Verde a maioria das empresas de base tecnológica e instituição de ensino superior não pagam pelas licenças dos softwares e isto tira o preço como vantagem de softwares *Open Source*, sabendo que as pessoas adquirem softwares Proprietário sem pagar licença.

1.6 Estrutura do trabalho

O presente trabalho encontra-se dividido em 3 Capítulos:

- O primeiro Capítulo abordamos o surgimento do software *Open Source*, sua evolução, principais pessoas que lutaram para o seu crescimento, alguns dos projetos de referência. Nesse mesmo Capítulo fala-se do conceito de software proprietário, o seu historial e também das principais licenças *Open Source*. As vantagens, desvantagens do software *Open Source* e seu impacto nos países como Cabo Verde também foram abordados;
- No segundo Capítulo exploramos as principais tecnologias *Open Source* existente, onde os seus historiais, suas características e índice de utilização em relação as suas alternativas proprietárias. Nesse caso para falar dessas tecnologias, elas foram separadas por categoria para uma melhor organização, onde foram selecionados alguns softwares;
- O terceiro Capítulo é o estudo de caso onde aplicamos dois questionários, um para empresas de base tecnológica e o outro para as instituições de ensino superior na Cidade da Praia, com o objetivo de saber quais as dificuldades que enfrentam quando utilizam um software *Open Source*, que critérios são mais considerados na escolha de

um determinado software, se dispõem ou não de técnicos capazes de resolver problemas referentes a softwares *Open Source* etc.

Capítulo 1: Software Open Source e software proprietário

Ao falar da informática é quase inevitável não falar de softwares, pois estes é que dão vida ao hardware; pode-se ter o melhor computador do mundo mas se não tiver nenhum software, ele fica inútil, isto porque não podemos utilizá-lo. Neste Capítulo, vamos abordar os dois tipos de softwares existentes, no contexto da disponibilidade do código fonte que são elas *Open Source* e *Closed Source* (software proprietário).

1 Open Source

De acordo com Meeker (2008) para entendermos o conceito do *Open Source* devemos entender antes de mais o que é código fonte e porque o acesso ao código fonte é importante.

Segundo Preesman (2006) *apud* Bressan e Filho (2007) um código fonte é um conjunto de instruções que executam uma função, por outras palavras, que controlam o funcionamento de um computador. Então, logo pode-se ver porque o acesso ao código fonte é muito importante, visto que é este que controla o funcionamento de um computador e ao ter acesso ao código fonte fica fácil fazer alterações das funcionalidades do programa.

1.1 Definição

“*Open Source* é uma evolução da prática de longa data de fornecer a fonte de software”(Weerawarana et al., 2004:p. 13), mas mais abaixo vamos ver que não é só fornecer o código fonte, segundo alguns autores como no caso do (Richard Stallman, 2007) .

De acordo com Lerner e Tirole (2005) *apud* Meireles (2009:p.1) “*Open Source* é um processo inovador de desenvolvimento de software que utiliza como fundamento a transparência de código fonte entre todos os pares.”

Segundo Richard Stallman (2007) um software por ser *Open Source* isto não quer dizer que este seja grátis mas que o código fonte é disponibilizado, de acordo com o mesmo autor é essa a grande diferença existente entre softwares *Open Sources* e Free software.

Grande quantidade de pessoas confundem softwares *Open Source* com os softwares livres ou “freeware”, mas os softwares *Open Source* distinguem-se dos chamados freeware devido ao facto de, ao contrário destes, os primeiros incluírem o código fonte do software, ou a indicação de onde e como o conseguir¹.

Open Source é todo software cujo código fonte esteja disponível para consulta (mesmo que sem possibilidade de livre modificação, ou de livre distribuição de versões modificadas) ”(Campos, 2006).

Segundo Fuggetta (2003:p.78) um software *Open Source* “ (...) é qualquer software cuja licença garante ao seu usuário liberdade relacionada à utilização, alteração e redistribuição. Seu aspeto fundamental é o facto do código fonte estar livremente disponível para ser lido, estudado ou modificado por qualquer pessoa”.

Campos (2006) afirma ainda que para um software ser considerado *Open Source*, sua licença tem que satisfazer dez condições, sem faltar nenhuma, ou seja pode satisfazer as 9 e faltar uma ela não é considerada *Open Source*.

Segundo Coar (2006) *apud* (I. B. O. da Rosa, 2010) as dez condições para que um Software seja considerado *Open Source* são:

1. Livre redistribuição - a licença não deverá restringir qualquer venda ou trespasse do software como componente de uma distribuição agregada do software contendo

¹Disponível em [<http://www.infopedia.pt>], consultada a 03 Julho de 2013

programas de várias fontes distintas. Nestes casos, a licença do software *Open Source* não deverá requerer qualquer pagamento pela venda efetuada;

2. Código fonte - o programa deverá incluir o código fonte e deverá permitir a distribuição tanto em código aberto ou de forma compilada. Caso alguma parte do código fonte não esteja disponível, dever-se-á publicar claramente, a forma de obtenção desse código fonte. Os custos não deverão ser superiores ao custo normal de reprodução, sendo o *download* através da Internet, o mecanismo preferencial para essa publicação;
3. Trabalhos derivados - a licença deverá permitir a realização de modificações, bem como a sua distribuição nos mesmos termos que a licença do software original;
4. Integridade do código fonte do autor - a licença do software *Open Source* deverá permitir explicitamente a distribuição do código fonte alterado, podendo entretanto requerer que seja publicado com um nome ou versão diferente do software original;
5. Não discriminação contra pessoas ou grupos - nenhuma pessoa ou grupo de pessoas deverá ser discriminado pela licença;
6. Não discriminação contra áreas ou empreendimentos - a licença não deverá impedir a utilização do software em nenhuma área de atividade ou em qualquer tipo de empreendimento;
7. Distribuição de licença - os direitos relativos ao programa deverá ser aplicada a todos aqueles a quem o programa for distribuído;
8. A licença não deverá ser específica de um softwares - não deverá impor restrições sobre outros softwares distribuídos conjuntamente com o software de código aberto;
9. Não restrição de outros softwares - a licença não deverá restringir outros softwares que são distribuídos juntamente com os que são *Open Source*;
10. A licença deverá ser tecnologicamente neutra - ao ser fornecida, a licença não deverá tecer considerações sobre nenhuma tecnologia.

Diante das várias definições apresentadas pelos autores acima, construímos uma definição que acreditamos ser uma junção das outras definições: software *Open Source* é todo software cujo código fonte encontra-se disponível de forma que todas as pessoas/utilizadores possam ter acesso a ele e respeitar as 10 condições citadas acima.

1.2 Historial

Segundo Weber (2004) "*no início*", não houve diferença significativa entre o que chamamos hoje em dia de hardware e de software, ou entre o usuário e o programador. Havia apenas o computador e as pessoas que trabalhavam com ele.

Sabino e Kon (2009) afirmam ainda que na década de 50, maioria das vezes ocorria uma venda casada entre hardware e software, pois os programas eram fortemente acoplados à arquitetura das máquinas em que eram executados. Nessa época, o foco das empresas era na venda do hardware, e não eram colocadas muitas restrições na utilização que as pessoas faziam do software. Pode-se dizer que na década de 50, não dava-se muita importância aos direitos autorais, sabendo que também as empresas não estavam com muita exigência no que diz respeito aos softwares, o software em si não era vendido.

Segundo Rosão et al. (2010) na década de 60, do século passado, a IBM que dominava o mercado tanto em termos de software como hardware, os programas eram vendidos em conjunto com o hardware, tal como se fizessem parte integrante do computador. Curiosamente, o código destes programas estava disponível e as pessoas faziam modificações e partilhavam-no entre si. Porém, pode-se perfeitamente dizer que os softwares eram *Open Source*.

Silva (2009) acrescenta ainda que o código fonte era distribuído porque era frequente a necessidade de consertar bugs e adicionar novas funcionalidades. Neste caso, estas ações já satisfaziam alguns dos critérios dum software *Open Source*.

Segundo Rosão et al. (2010) já a partir da década de 70 as coisas tomaram um novo rumo, visto que o software começou a ser encarado como algo com valor intrínseco e, consequentemente, como uma oportunidade de negócio. Desta forma, a IBM resolveu vendê-lo separadamente do hardware e começou a restringir o acesso ao código fonte do mesmo, algo que se tornou um padrão ainda em vigor hoje, por muitas empresas, cerca de 40 anos

depois. Iniciou o que até hoje está em vigor em muitas empresas que é o software proprietário.

O mesmo autor (*idem*) afirma ainda que no fim da década de 70 muitas outras empresas seguiram o exemplo da IBM. Porém, a IBM não foi o único que não estava de acordo com as partilhas e modificações do seu código fonte.

Silva (2009) afirma que em 1976 Bill Gates escreveu a Carta Aberta aos Hobistas, dizendo que o que os *hackers*² chamavam de “compartilhamento” era, para ele, “roubo”.

(I. B. O. da Rosa, 2010) afirma que no início da década de 80, Unix tornou-se restrito apenas aos que tivessem pago as licenças para utilizá-lo. Contudo, já na década de 80 era normal a utilização da licença nos softwares produzidos pelas empresas de desenvolvimento.

Rosão et al. (2010) afirma ainda que só em 1984 nasceu um dos projetos mais importantes da história do software *Open Source*: GNU concebido por Richard Stallman, com o objetivo de criar um sistema operativo gratuito, *Open Source* e colaborativo e que assegurasse aos seus utilizadores quatro liberdades fundamentais.

Em 27 de Setembro de 1983, Richard Stallman postou uma mensagem no net.unix-wizards e net.usoft newsgroups com o assunto “*New Unix implementation*”³. Nessa mensagem, ele informa que está começando a escrever um sistema compatível com UNIX chamado GNU e que ele será dado a todas as pessoas interessadas. Ele cita alguns componentes que seriam incluídos, tais como núcleo do sistema operativo, compilador C e editor de texto, e propõe algumas melhorias em relação aos sistemas UNIX existentes na época. Ele também explica na mensagem o motivo dele precisar escrever o GNU: segundo seus princípios, se ele gosta de um programa ele precisa compartilhá-lo com outras pessoas que também gostem dele. Para continuar utilizando computadores sem violar seus princípios, ele decidiu criar um conjunto suficiente de software *Open Source* para que ele pudesse prosseguir sem utilizar qualquer software que não fosse *Open Source*. Para finalizar a mensagem, ele pede contribuições na forma de máquinas, ajuda para escrever o software e dinheiro (Sabino e Kon, 2009:p.2).

Stallman, como vê-se no trecho do texto acima foi um dos grandes lutadores para o surgimento do movimento *Open Source*; muitos autores o consideram o pai do *Open Source*.

Stallman (2007) afirma que em 1984 eles iniciaram o desenvolvimento do sistema operativo free GNU para que eles pudessem evitar os sistemas operativo proprietários que negam liberdade aos seus usuários.

² *Hacker* é um indivíduo que se dedica, com intensidade incomum, a conhecer e modificar os aspetos mais internos de dispositivos, programas e redes de computadores.

³ Que ao traduzir para português fica “*Nova implementação do Unix*”.

Entre 1989 e 1990, um estudante de ciência da computação de graduação da Universidade de Helsinki, Linus Torvalds, iniciou o projeto que completou o circuito e totalmente energizada a visão de software *Open Source* (Moglen, 2013).

O resultado do trabalho de Torvalds foi lançado na internet, em 1991 (...) surgiu o Linux uma das referências do *Open Source* segundo referencia (Moglen, 2013).

“Em 1992, o kernel Linux foi incorporado aos programas GNU, gerando um sistema operativo *Open Source* completo, que recebeu o nome de GNU/Linux, embora hoje seja mais conhecido como apenas Linux” (Bacic, 2003:p.13).

Mas, até aqui se deparamos não surgiu nenhum conceito da *Open Source*; em suma, passaram décadas em que muitas empresas disponibilizavam os seus códigos fontes para o público, mas o conceito não existia ainda.

De acordo com a OSI o conceito *Open Source* surgiu no ano 1998 numa reunião em Califórnia.

Segundo Weber (2004) Netscape chocou o mundo da computação, quando, em Janeiro de 1998, eles anunciaram que iriam liberar o código-fonte de seu navegador.

A década de 90 foi onde o *Open Source* teve o seu surgimento como conceito e também houve grande lançamento de softwares *Open Source* como por exemplo Linux e Netscap ficou *Open Source* e entre outros.

(I. B. O. da Rosa, 2010) indica ainda um outro evento na década de 90 que deu um contributo muito importante para o movimento *Open Source*, quando Bill Jolitz concluiu o desenvolvimento do 386BSD, uma versão do Unix fora das restrições de licenciamento do Unix da AT&T.

Meeker (2008) afirma que os códigos do Netscap quando libertados deram origem ao Firefox, um dos projetos da Mozilla de muito sucesso.

O mesmo autor (*idem*) afirma que *Open Source Initiative* (OSI) foi formado com a missão de harmonizar *Open Source* e software proprietário. O que não está acontecendo como o

planeado. No entanto pode se concluir que *Open Source* surgiu como conceito em 1998 e os critérios para que um software seja *Open Source* também no mesmo ano.

1.3 Open Source e Free software (Diferença)

É muito comum encontrarmos pessoas a fazer confusão entre os dois conceitos, visto que chamam de *Open Source* alguns softwares pelo facto deste ser grátis.

“O *Free* software é todo software que oferece ao seu usuário a liberdade de utilizá-lo para seu proveito, executando os programas para quaisquer propósitos, copiando, distribuindo, estudando seu funcionamento, modificando, aperfeiçoando e personalizando seu código fonte, sem precisar pedir autorização para outrem, de modo que toda a sociedade se beneficie” (Lamas, 2004) *apud* (Bressan e Filho, 2007:p.4). Segundo Meeker (2008) para um software ser considerado Free este tem de satisfazer 4 condições:

- A liberdade de executar o softwares para qualquer propósito;
- A liberdade de estudar como o software funciona, e adaptá-lo às suas necessidades;
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo;
- A liberdade de melhorar o software, e liberar as suas melhorias ao público, de modo a beneficiar toda a comunidade;

Segundo o mesmo autor (*idem*) a liberdade 1 e 3 precisam do código fonte.

Porém, o simples facto do software ser *Open Source* não garante absolutamente nada sobre a sua distribuição, modificação e comercialização. Na verdade, o termo “*Open Source*” somente afirma que qualquer pessoa pode ter acesso ao código fonte do software e respeitar as 10 condições citadas acima, mas que seu desenvolvedor é que determina as suas condições de utilização (Gugik, 2009).

Segundo Richard Stallman (2007) acrescenta ainda que todos softwares Free se qualificaria como software *Open Source*, mas nem todos softwares *Open Source* se qualificaria como software free visto que de acordo com o mesmo autor (*idem*) tem vários softwares *Open Source* que possuem licenças muito restritivas.

Sendo assim, pode-se dizer de uma forma geral que principal diferença é o facto de *Free* software oferecer total liberdade enquanto que no *Open Source* as liberdades oferecidas são controladas pelo dono do software; por exemplo se a Microsoft disponibilizar o código fonte do Windows e restringir a liberdade de modificação este é considerado um software *Open Source*, uma vez que restringe a liberdade de modificação oferecida pelos softwares *Free*.

2 Software proprietário

2.1 Definição

O termo proprietário é utilizado por desenvolvedores de softwares para designar os softwares que não são *Open Source* (Meeker, 2008).

De acordo com a história do software *Open Source*, no início os softwares eram vendidos juntamente com os hardwares e não tinha grandes restrições no que se refere ao software e nem havia distinção alguma.

Com o passar dos anos a indústria de software iniciou sua fase decrescimento e, com o aumento incrível de performance dos computadores nos anos 80, os ambientes de desenvolvimento de software se multiplicaram. As empresas começaram a desenvolver softwares aplicativos e a comercializá-los. A essa forma de distribuição de software denominou-se “software Proprietário”, ou seja, aquele software cuja cópia, redistribuição ou modificação são em alguma medida proibidos pelo seu criador ou distribuidor (Farias, 2011:p.29).

Um software proprietário é um programa de computador que possui o seu código fonte fechado, onde somente um indivíduo ou um grupo especializado da empresa fabricante tem o controlo sobre seu código e consequentemente sobre suas funções, melhoramentos e correções (Bressan e Filho, 2007).

No entanto, há autores que contradizem parte da afirmação acima, como é o caso de (I. B. O. da Rosa, 2010) que afirma que “apesar da maioria dos softwares proprietários ser de código fechado, podem existir casos de disponibilização do código fonte. Entretanto, isso normalmente é feito mediante licenças específicas podendo requerer igualmente a assinatura de um acordo de não disponibilização desse mesmo código fonte a terceiros”.

Ou melhor, pode-se dizer que um software proprietário pode ser *Open Source* perante acordo entre o fornecedor e o cliente.

Segundo LINFO (2005) nos softwares proprietários quase sempre há grandes restrições à sua utilização, e seu código fonte é quase sempre mantida em segredo. Na opinião de Ferraz (2002:p.10) “softwares proprietários são aqueles que não oferecem qualquer uma das três liberdades fundamentais (utilização, modificação ou distribuição) ”, por outras palavras, um software é considerado não proprietário se negar uma das quatro liberdades.

Essas restrições dão certas características aos softwares proprietários que segundo Bressan e Filho (2007) tal configuração de propriedade e o carácter pago de sua licença conferem-lhe um carácter comercial; esse intento aproxima-se de uma certa resistência social quando se constata que poucas corporações dominam a comercialização de tais softwares à escala mundial.

Alguns observadores da indústria acham que o papel do software proprietário vai diminuir no futuro por causa da crescente concorrência de software *Open Source*. Este ponto de vista sustenta que o software *Open Source* vai, eventualmente, vir a dominar os sistemas operativo e os principais programas da aplicação. Software proprietário continuará forte em alguns nichos de mercado, principalmente para aplicações de negócios e técnica para que a demanda é relativamente pequena ou especializada e para que os usuários estarão dispostos a pagar preços relativamente altos (LINFO, 2005).

Pode-se encontrar alguns exemplos de softwares *Open Source* que já são concorrentes fortes à suas alternativas proprietário, como o caso do Servidor Apache, sendo o servidor *Web* mais utilizado a nível mundial; o GIMP é muito utilizado tendo como alternativa proprietária o Photoshop, que é muito utilizado e mais popular que o GIMP mesmo sendo pago; o Mozilla Firefox que é um *browser* muito utilizado estando neste momento em 3º lugar entre os 5 *browsers* mais utilizados, mas este já estivera em posições melhores no passado.

Com as definições citadas por alguns autores, cada um com o seu ponto de vista, construímos uma definição que achamos ser mais realista: software proprietário é todo programa que não disponibiliza o seu código fonte e também não oferece uma ou nenhuma das 10 condições para um software ser considerado *Open Source* citada por (I. B. O. da Rosa, 2010) na página 20.

2.2 Historial

Como vê-se no historial do software *Open Source*, no início, tudo era *Open Source*, por outras palavras, quando um cliente adquirir um software, o código fonte era disponibilizado e sem grandes restrições no que se refere à alteração, divulgação, cópia, entre outros.

Segundo Sabino e Kon (2009), na década de 1970 a situação começou a se modificar. Algumas empresas, como a Microsoft, não estavam satisfeitas com a forma como seus programas eram redistribuídos sem que a empresa recebesse nenhum crédito pelas cópias. Com isso, pode-se afirmar que o software proprietário teve o seu início na década de 70.

Posteriormente, com a maior disseminação dos computadores e o crescimento da indústria de Tecnologia da Informação, o software começou a ser vendido separadamente da máquina. Gerou uma indústria bilionária, e as empresas do sector começaram a buscar mecanismos de proteção de propriedade intelectual, como direitos autorais (copyrights) e patentes, para se defender da acirrada concorrência e garantir suas vantagens competitivas. Neste momento, o código-fonte passou a ser protegido, pois ele é o próprio conhecimento do programa (Taurion, 2004).

Ou seja, com o aumento do número de computadores repararam que podiam ganhar com a proteção dos softwares, então a maioria das empresas de desenvolvimento protegeu o código fonte dos seus softwares.

Taurion (2004) afirma ainda que o propósito do surgimento do software proprietário foi para preencher uma necessidade legítima do mercado. Ele ainda afirma que sem o incentivo financeiro a indústria do software não teria chegado aonde chegou.

Por outras palavras, mudou-se a forma da comercialização dos softwares, visto que antes o software era vendido em conjunto com o hardware, o que contribuiu para o crescimento de muitas empresas, como é o caso da Microsoft.

Segundo Iwata (2009), durante a década de 80 o MS-DOS foi o sistema operativo mais utilizado para as plataformas compatíveis com o PC, o MS-DOS.

No entanto, podemos ver que não é de hoje que os sistemas operativos da Microsoft estão a dominar o mercado.

Em 1985 a Microsoft lançou um ambiente de sistema operativo com o nome do Windows como um suplemento para MS-DOS (Iwata, 2009).

Segundo Iwata (2009), o Windows está instalado em maioria dos computadores pessoais (PC), e um dos grandes motivos do Windows possuir essa enorme parcela de mercado foi dominância do MS-DOS nos primórdios da implantação dos PCs e computadores compatíveis (clones do IBM PC). No entanto uma pequena questão “Windows não é gratuito porque é que está instalado em maioria dos computadores?”, uma possível hipótese é a sua dominação do mercado com versão anteriores e uma outra hipótese é a questão de o Windows ser muito pirateado.

3 Licença Open Source

Para definir o modo como as pessoas utilizam um determinado software é utilizado uma licença de software.

Na opinião de Lindberg (2008) a palavra "licença" tem, e teve por centenas de anos, um significado técnico específico na lei de propriedade. A licença é uma autorização unilateral de utilizar propriedade alheia.

Uma das preocupações mais comuns sobre o *Open Source* está relacionada com as licenças, patentes e direitos autorais (Weerawarana et al., 2004). O mesmo autor (*idem*) afirma que a *Open Source Initiative* certificou cerca de 40 licenças *Open Source*, mas neste trabalho só vai ser abordado alguns dos mais populares.

Bacic (2003) afirma que ao longo do tempo diversas licenças surgiram, apresentando de uma forma geral as seguintes características:

- Proteger a identidade do autor;
- Distribuir os códigos fontes;
- Qualquer trabalho que envolva parte do software deve citar o autor;
- Instalação do software em número ilimitado de computadores;
- Não restringir a venda do software.

No caso das características acima apresentadas, referem-se a uma licença dos softwares *Open Source*.

Com isso, pode-se perfeitamente dizer que é a licença que determina tudo o que se pode fazer com um determinado software.

De acordo com Meeker (2008) para melhor efetuar os propósitos da *Open Source*, licenciados precisam saber:

- Quais são os componentes *Open Source*;
- Se os componentes são modificados ou não modificados;
- Onde o código fonte pode ser adquirido;
- O que governa a licença de utilização.

No entanto, antes de estudar alguma das licenças dos softwares *Open Source* primeiro vai ser apresentado dois conceitos o copyleft e o copyright.

Copyright

Direito⁴ autoral, direitos autorais ou direitos de autor são as denominações empregadas em referência ao leque de direitos dos autores sobre suas obras intelectuais, sejam estas literárias, artísticas ou científicas. Segundo a doutrina jurídica clássica, nesse leque encontram-se direitos de natureza pessoal e patrimonial, também denominados, respetivamente, direitos morais e direitos patrimonial.

De acordo com Lindberg (2008), o copyright é utilizado para proteger tudo e qualquer criação que se encontra escrito ou gravado em algum lugar; ele também acrescenta que o copyright demora de 90 a 150 anos dependendo da circunstância.

O mesmo autor (*idem*) afirma ainda que o que difere copyright de patentes, é que o copyright não proíbe a mesma ideia e patente proíbe a mesma ideia, por exemplo se uma pessoa

⁴ Disponível em [https://pt.wikipedia.org/wiki/Direito_autoral], consultado em 16/07/2013

construir qualquer coisa e patenteá-la, ninguém tem o direito de criar algo idêntico, mas no caso de copyright outras pessoas podem construir algo com a mesma ideia.



Figura 1: Símbolo do copyright (Wikimedia, 2013b).

Copyleft

Segundo (R. L. Silva, 2005), o software *Open Source* faz utilização de um copyright para consolidar a proteção, mas utiliza o termo copyleft (algo como esquerdo autoral), que nada mais é que uma paródia para contrapor o atual conceito de copyright. Copyleft é uma forma de utilizar a legislação de proteção dos direitos autorais com o objetivo de retirar barreiras à utilização, difusão e modificação de uma obra criativa devido à aplicação clássica das normas de propriedade intelectual, exigindo que as mesmas liberdades sejam preservadas em versões modificadas (“Wikipédia,” 2013e). Segundo Sabino e Kon (2009:p.17) “a licença que deu origem à ideia de copyleft foi a General Public License, ou GPL da Free Software Foundation.”

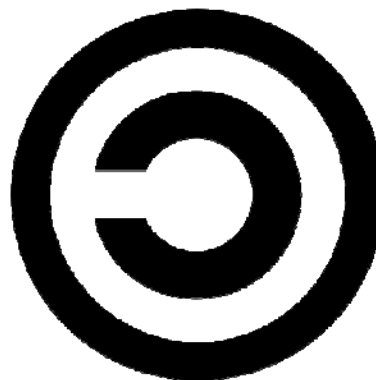


Figura 2: Símbolo de copyleft. (Wikimedia, 2013a)

O (Richard Stallman, 2007) define ainda o copyleft de uma forma mais simplificada e como ela se-originou. Pode-se ler nos trechos do texto em abaixo para que possamos entender da melhor forma:

O modo mais simples de tornar um programa *Open Source* é colocá-lo em domínio público, sem copyright. Isto permite que as pessoas compartilhem o programa e suas melhorias, se elas estiverem dispostas a tal. Mas isto também permite que pessoas não-cooperativas transformem o programa em software proprietário. Eles podem fazer modificações, poucas ou muitas, e distribuir o resultado como um software proprietário. As pessoas que receberem esta forma modificada do programa não têm a liberdade que o autor original havia-lhes dado; o intermediário eliminou estas liberdades.

O copyleft diz que qualquer um que distribui o software, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa. O copyleft garante que todos os utilizadores têm liberdade.

Por outras palavras, o copyleft permite a todos os utilizadores a execução, o estudo e a modificação dos softwares, mas a sua divulgação implica que este tem de estar sob a mesma licença que o software original. Por exemplo, no caso de um software estar sob a licença Eclipse Public License, que é uma licença copyleft, qualquer pessoa que o altere, para divulga-lo deve estar sob a mesma licença.

No quadro da figura abaixo pode-se ver algumas licenças *Open Source*.

Licenças	Hereditário	Criador/Autor	Projectos	Comentário
Apache Licence	Não	Apache Foundation	Ant, Tomcat, Xerxes	Licença permissiva, mas termos muito mais detalhadas do que BSD ou MIT, ou Apache 1.0 ou 1.1.
BSD	Não	FreeBSD		Modelo de licença de muitas variantes em uso. Variantes principais são o "3-clause" e variantes "2-clause" versões anteriores contidas e cláusula de publicidade.
Common Development and Distribution License	Sim	Sun	Open Solaris Netbeans	Recente adição à lista OSI, com base na MPL. Sucessor Sun Public License
Eclipse Public License	Sim	Eclipse Foundation	Eclipse project	Variante de CPL
GPL	Sim	FSF	Linux, GNU	Licença mais utilizado
MIT license	Não		X Window System, Putty	Licença permissiva mais popular, depois de BSD. Também chamado de X ou de
Mozilla Public License	Sim	Mozilla Foundation	Firefox, Thunderbird	

Figura 3: Licenças Open Source, projetos e autores, adaptado de Meeker (2008)

Para além dessas licenças existem muitas outras que não foram referidas no quadro acima, bem como os seus criadores.

Apache

De acordo com (“Wikipédia,” 2013e), a licença Apache é para software *Open Source* de autoria da Apache Software Foundation (ASF). Todo software produzido pela ASF ou qualquer um dos seus projetos é licenciado de acordo com os termos da licença Apache. Alguns projetos não pertencentes à ASF também utilizam esta licença. Algumas versões da licença Apache exige a inclusão do aviso de copyright, mas não é uma licença copyleft; ela permite a utilização e distribuição do código fonte tanto no software *Open Source*, como no proprietário. Todavia, há projeto de grande influência no mundo, que está sob esta licença como no caso do Servidor Apache que segundo as estatísticas da Netcraft⁵ tem 52% dos *sites* atuais. Pode-se reparar tabela da figura 3 que também para além de Servidor Apache há também outros projetos como Tomcat, Xerxes etc. que utilizam a licença Apache.

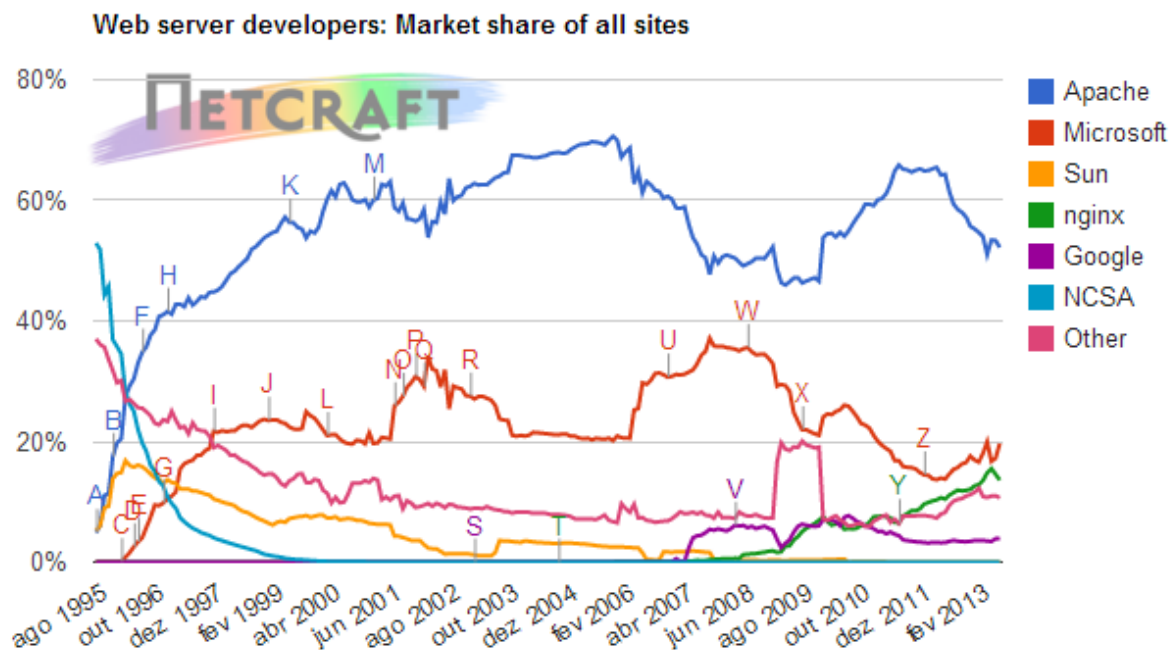


Figura 4: Parcela de mercado dos principais servidores Web (Netcraft, 2013)

⁵ Netcraft oferece análise de quota de mercado para mercados de hospedagem de *sites* e de servidores web, incluindo detecção do sistema operativo e do servidor web.

O gráfico da figura 4 mostra-nos uma diferença entre o servidor Apache e o principal rival da Microsoft no que se refere a popularidade, que representa 20% dos *sites* atuais. Em Agosto 1995 o mercado de servidores Web era liderado pelo servidor NCSA, mas este foi destronado pelo servidor *Open Source* o Apache logo no ano seguinte.

BSD

A licença de BSD⁶ (Berkeley Software Distribution) é uma licença lançada pela Universidade da Califórnia, diferente de outras licenças copyleft como no caso do GPL; ela não é copyleft, no entanto permite que parte de software lançado sobre esta licença seja incluída nos softwares proprietários. BSD permite a utilização livre dos seus softwares, distribuição de cópias sem nenhuma restrição no número de cópias, mas também não responsabilizam por qualquer danos ou inconveniências causadas pelo programa. Esta licença não impõe muitas restrições, mas no caso de uma pessoa querer fazer uma nova distribuição de softwares que contenham todo ou parte do software, este deve reconhecer em seus materiais de propaganda que esses softwares contêm software desenvolvido pela UC Berkeley e seus colaboradores. Mas segundo Laurent (2008) isto foi antes de 1999, que a licença BSD continha a seguinte cláusula:

"3. Todos os materiais de publicidade que mencionem funções ou a utilização deste software devem exibir a seguinte declaração:

Este software inclui software desenvolvido pela Universidade de Califórnia, Berkeley e seus colaboradores. "

Em suma, os utilizadores têm de dar crédito a desenvolvedores originais para poderem usufruir de softwares sob a licença BSD. “Parece razoável e natural que o criador, tendo rendido o direito exclusivo de explorar comercialmente uma obra, deve receber o crédito, não só no reconhecimento dos direitos, mas na publicidade também” (Laurent, 2008:p.16). Mas o mesmo autor (*idem*) afirma que essa cláusula não desafia as premissas essenciais do *Open Source*, uma vez que não limita o âmbito da utilização do software de qualquer maneira direta.

GPL

⁶Disponível em [<http://www.freebsd.org/copyright/license.html>], consultado a 16 de Julho de 2013

O GPL é a licença de código aberto mais utilizado (Meeker, 2008). Segundo Lindberg (2008:p.223) “muitas das perguntas mais difíceis em software free e *Open Source* giram em torno do GPL. O GPL tem um monte de coisas que vão para ele: é a única licença de software de código aberto mais comum; ele reuniu um grande e vibrante comunidade de desenvolvedores, e *hackers* brilhante (...)”. Esta licença é aplicada apenas em softwares *Open Source*, por isso que se diz que é uma licença de *Open Source*.⁷ Ele é copyleft isto porque uma pessoa pode vendê-lo ou redistribuí-lo gratuitamente mas tem de passar essa liberdade adiante. Desenvolvedores que utilizam o GNU GPL, para proteger seus direitos são precisos dois passos:

- Afirmar direitos autorais sobre o software;
- Oferecer licença que dá ao utilizador a permissão legal para copiar, distribuir e / ou modificá-lo.

Segundo Lindberg (2008) o conceito de copyleft embutido no GPL é uma maneira mais revolucionária de pensar sobre a propriedade intelectual, que segundo o mesmo é vital e importante.

Para a proteção dos desenvolvedores e autores, o GPL explica claramente que não há nenhuma garantia para este software livre. GPL requer que versões modificadas sejam marcadas como modificadas, de modo que os seus problemas não sejam atribuídos erroneamente a autores de versões anteriores.

Na licença GPL há uma exceção especial que permite códigos proprietários sejam combinados com os mesmos ficheiros executáveis (ligação, dinâmica ou estática) para código coberto por esta licença (Meeker, 2008).

Se repararmos na figura 3, muitos projetos de grande importância no mundo da *Open Source*, utilizam a licença GPL, como no caso de Linux, e Laurent (2008) cita outros softwares que utilizam esta licença como GNU Emacs, compilador GNU C, etc.

MIT license

⁷Disponível em [<http://www.gnu.org/licenses/gpl.html>], consultado a 16 de Julho de 2013

A licença MIT⁸, também chamada de “licença X” ou de “licença X11”, criada pelo Massachusetts Institute of Technology, é uma licença não copyleft utilizada em software *Open Source*, isto é, por ser não copyleft ela permite a reutilização de software licenciado em softwares *Open Source* ou proprietários.

Isto quer dizer que um software proveniente de um outro software que tem esta licença não é obrigado a ser da mesma licença, visto que ela não é copyleft.

⁹Não há copyright no texto da licença MIT, por isso, outros grupos podem modificar a licença para atender as suas necessidades. Por exemplo, a Free Software Foundation utiliza a licença MIT para a sua biblioteca ncurses¹⁰, com a adição do seguinte texto:

- Com exceção do contido no presente aviso, o nome do (s) dos detentores de direitos autorais acima anunciados;
- Não deve ser utilizado em publicidade ou de outra forma para promover a venda, utilização ou em outras negociações, este mesmo software sem autorização prévia por escrito.

Mozilla Public License

Segundo Laurent (2008), a Mozilla Public License surgiu de uma mistura de duas licenças, que de acordo com a mesma fonte essas duas licenças que deram origem ao MPL são GLP e BSD.

¹¹Esta é uma licença de software *Open Source* que não exige copyleft forte; ao contrário da licença GPL, ela tem algumas restrições complexas que a tornam incompatível com a GNU GPL. Por outras palavras, um módulo coberto pelo GPL e um módulo coberto pela MPL não podem ser *linkeditados*¹² conjuntamente. No entanto, Laurent (2008) reforça ainda que um código/software coberto pela licença MPL pode ser utilizado em combinação com outros softwares que possuem outras licenças, o que não é permitido pela licença GPL e é permitido pela licença BSD; este é um dos fatores que o tornam incompatível com a licença

⁸ Disponível em [http://pt.wikipedia.org/wiki/Licen%C3%A7a_MIT], consultado a 16 de Julho de 2013

⁹ Disponível em [http://pt.wikipedia.org/wiki/Licen%C3%A7a_MIT], consultado a 17 de Julho de 2013

¹⁰ É uma biblioteca capaz de dar funcionalidade a terminais a base de texto. Com ela os seus programas poderão: ter janelas; utilizar cores; ter suporte a *mouse*; utilizar teclas de função do teclado.

¹¹ Disponível em [<http://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>], consultado a 16 de Julho de 2013.

¹² *Linkeditado* significa ligar programas no momento da compilação.

GPL; segundo o mesmo autor (*idem*) a MPL estabelece uma espécie de meio-termo entre as duas licenças (GPL e BSD).

Common Development and Distribution License

A Common Development and Distribution License (CDDL) ¹³ É uma licença *Open Source* aprovada pela OSI, que é utilizado pelo projeto OpenDS, bem como uma série de outros projetos de código aberto, incluindo OpenSolaris e OpenSSO. A CDDL é baseada no Mozilla Public License (MPL) e inclui uma cláusula de concessão da patente para que qualquer tecnologia coberto por patentes seja concedida a outros projetos utilizando o código.

Segundo a tabela da figura 3 pode se ver que a Common Development and Distribution License é uma licença copyleft.

Eclipse Public License

O Eclipse Public License ¹⁴ é uma licença *Open Source* de copyleft fraco, utilizado pela corporação sem fins lucrativos, no entanto o Eclipse Foundation em seus projetos de software utiliza esta licença. Não é amplamente utilizado fora da Eclipse Foundation. Porém, a popularidade do software desenvolvido pela Eclipse Foundation, como o Eclipse IDE, Rich Client Platform, ferramentas de Business Intelligence e entre outros projetos torna esta licença muito utilizado. Lindberg (2008) acrescenta ainda que o Eclipse Public License faz parte da lista das 9 licenças mais utilizadas ou que tenham uma comunidade mais forte.

Segundo o mesmo autor (*idem*) esta licença também faz parte das listas de licenças Open Sources que não são compatíveis com o GNU GPL.

3.1 Problemas relacionados com as licenças Open Sources

Imaginemos que temos um jantar na nossa casa e convidamos familiares e amigos com gosto para alimentos diversificados. Neste caso torna-se mais difícil satisfazer o desejo de todos e seria mais fácil se todos tivessem o mesmo gosto.

¹³ Disponível em [<http://docs.oracle.com/cd/E19450-01/820-6173/def-common-development-and-distribution-license.html>], consultado a 17 de Julho de 2013.

¹⁴ Disponível em [<http://www.oss-watch.ac.uk/resources/epl>], consultado a 17 de Julho de 2013.

Segundo (Meeker, 2008) é exatamente o mesmo que acontece no mundo dos softwares, visto que todas as licenças contêm suas cláusulas e restrições exclusivas o que dificultam na combinação de softwares.

No entanto, pode-se verificar no quadro da figura 3 que há muitas licenças copyleft e outros não copyleft; e como se vê na página 32, que um software cuja licença é copyleft todo o trabalho proveniente dele tem de estar sob a mesma licença, nas licenças não copyleft isto não se aplica.

Pode-se conferir a cláusula 6 do GPL que é uma licença *Open Source* (Meeker, 2008:p.57):

6. Cada vez que você redistribuir o Programa (ou qualquer trabalho baseado no Programa), o receptor automaticamente recebe uma licença do licenciante original, para copiar, distribuir ou modificar o programa sujeito a esses termos e condições. Você não pode impor quaisquer outras restrições ao exercício dos direitos aqui concedidos aos destinatários.

Com essa cláusula a licença GPL proíbe qualquer utilizador final de negar alguma das liberdades oferecidas pela licença GPL como de copiar, distribuir e modificar, ou melhor dizendo, essa cláusula proíbe qualquer utilizador de tornar proprietário qualquer software sob esta licença, o que uma licença como BSD que não é copyleft já não proíbe a comercialização do trabalho proveniente, esse facto pode ser um problema se empresas querem desenvolver software que utilizam parte do software coberto pela licença GPL e outra parte coberto pela licença BSD.

4 Open Source versus Closed Source

Hoje em dia, há muitos defensores dos softwares *Open Source*, comunidades de discussão e desenvolvimentos, mas não pode-se negar a forte presença de softwares proprietários. Há pessoas que optam por softwares *Open Source* porque não gostam da filosofia dos softwares proprietário, outros porque dizem que softwares *Open Source* são mais seguros, outros porque têm gostam da política da Microsoft que segundo alguns autores é a referência dos softwares proprietário; há também pessoas que utilizam os softwares *Open Source* por serem mais baratos ou em maioria dos casos grátis. No entanto, escolhemos abaixo alguns critérios a serem analisados nos softwares *Open Source* e nos softwares proprietários fazendo assim uma

pequena comparação, apresentando tabelas e figuras de modo a ficar mais elucidado a diferença entre esses dois tipos de softwares e a sua utilização.

4.1 Custo

Segundo Michler (2005) não há mérito no argumento de que o software *Open Source* apresenta uma série de benefícios para o utilizador do software. Em particular, uma das principais vantagens atribuídas aos softwares *Open Source* é que é significativamente menos dispendioso do que o Software proprietário. Enquanto que os softwares proprietários maioria são pagas, mas temos alguns exemplos de softwares proprietários grátis como no caso de Google Chrome, iTunes, Teamviewer etc.

“Os produtores de softwares proprietários são incentivados a disponibilizarem novas versões melhoradas do software, de vez em quando de forma que os utilizadores possam ser obrigados a comprar as novas versões” (Dalle et al., 2001) *apud* (RosaB, 2010:p.104). Há outros autores como Ferraz (2002) que reforçam a ideia de que as empresas produtoras de softwares proprietários forcem os consumidores a atualizar para uma versão mais recente:

É importante lembrar que muitos fabricantes de softwares proprietários deixam de suportar seus antigos sistemas, forçando um *upgrade*. No caso da Microsoft, por exemplo, qualquer empresa que ainda utilize Windows 95, NT, 98 e ME, é fortemente incentivada a adquirir versões mais novas do Windows:

“Microsoft said customers need to upgrade to the latest versions of Windows to get stronger security. Customers' continued reliance on earlier versions of Windows, rather than the current Windows 2000 and Windows XP, is slowing down efforts to secure the global computing infrastructure, Craig Mundie, Microsoft chief technical officer, said in an address at the company's campus in Mountain View, Calif. He said it's impossible to retrofit earlier versions of Windows to make them secure” (Ferraz, 2002:p.49).

O facto de essas empresas lançarem atualizações pagas na maioria dos casos, afeta muito na escolha dos softwares, isto porque a maioria das suas alternativas *Open Source* lançam atualizações grátis.

4.2 Serviço e suporte

Um dos critérios importantes na escolha de um software é o suporte, visto que as empresas gastam muito dinheiro com o suporte e os utilizadores estão sempre a pesquisar sobre o suporte segundo a figura 5.

Segundo Steward (2011) o facto de o código fonte dos softwares *Open Source* estarem disponível qualquer empresa de desenvolvimento de software pode fornecer suporte para utilizadores dos softwares *Open Source*. Além disso, os clientes podem dar suporte a si mesmo. O que dificilmente acontece com os softwares proprietários, que na maioria dos casos o suporte é fornecido apenas pela mesma empresa.

Pode-se conferir no gráfico seguinte da Google Trends que as pessoas pesquisam muito a respeito de segurança e suporte da *Open Source*, isto quer dizer que é um facto que preocupa os utilizadores.

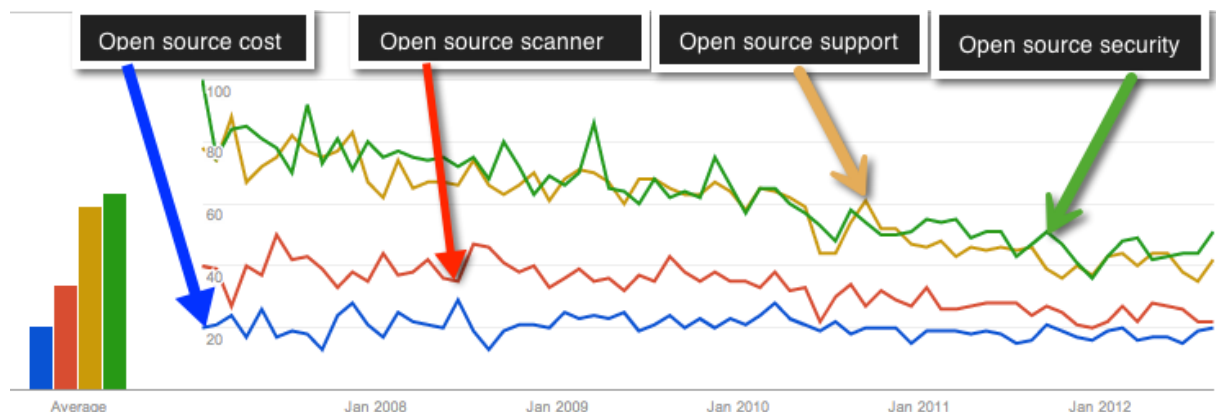


Figura 5: Índice de pesquisa de custos e suporte dos softwares Open Sources (Mandelbaum, 2012)

Pode-se verificar no gráfico da figura acima que um dos assuntos mais pesquisados sobre a *Open Source* é o suporte e segurança; isto mostra-nos claramente que segurança e suporte são principais preocupações dos utilizadores no que diz respeito ao *Open Source*.

4.3 Inovação

O conceito de inovação é bastante variado, dependendo, principalmente, da sua aplicação. “De forma sucinta, a inovação é a exploração com sucesso de novas ideias”(“Inventta,” 2013).

Segundo Ferraz (2002) a liberdade de utilização, modificação e distribuição favorecem a inovação. O software *Open Source* permite que pessoas ou empresas retomem o trabalho a partir do ponto onde outra pessoa ou empresa parou, ou continue o trabalho de maneiras imprevistas. Meireles (2009) reforça a ideia de que software *Open Source* traz inovação afirmando que “*Open Source* é um processo inovador de desenvolvimento de software que utiliza como base a transparência de código fonte entre todos os pares.”

Isto quer dizer que fica muito mais fácil surgir um software inovador a partir de um outro software *Open Source*, visto que neste, o código fonte encontra-se disponível, o que proporciona aos utilizadores a oportunidade de estudar e desenvolver uma melhoria ou até mesmo um novo software com base no anterior, o que não ocorre nos softwares proprietário se seguirmos o raciocínio da disponibilização do código fonte.

Inovação Fechada	Inovação Aberta
As pessoas inteligentes na área trabalham para nós.	Precisamos trabalhar com pessoas inteligentes dentro fora da empresa.
Se descobrimos nós mesmo, vamos leva-lo ao mercado primeiro.	Não temos que criar algo para lucrar com ele.
A empresa que levar uma inovação ao mercado primeiro vai ganhar.	Construindo um modelo de negocia melhor, é melhor do que chegar ao mercado primeiro.
Se criamos mais e melhores ideias na empresa vamos ganhar.	Se fizermos utilização de ideias internas e externas vamos ganhar.
Nós devemos controlar as nossas máquinas de modo que os nossos concorrentes não lucram com as nossas ideias.	*****

Tabela 1: Comparação entre filosofia OS e SP, adaptado de Engelfriet (2003).

Analisando a tabela 1, pode-se reparar que softwares *Open Source* e software proprietário têm uma filosofia totalmente diferente no que diz respeito à inovação, pois no *Open Source* trabalha-se com pessoas dentro e fora da empresa, enquanto que no proprietário trabalha só pessoas dentro da empresa. Segundo Engelfriet (2003) a *Open Innovation* se resume em ampliar os próprios fundamentos do mercado, criando novas oportunidades para todas as empresas do sector.

4.4 Usabilidade

De acordo com Nielsen (2008) *apud* Santos (2012:p.13) “usabilidade é um atributo de qualidade relacionado à facilidade da utilização de algo. Mais especificamente, refere-se à rapidez com que os usuários podem aprender a usar alguma coisa, à eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-la”.

“A necessidade de usabilidade tem sido cada vez mais percebida e a interface é um caminho fundamental para diferenciar softwares num mercado altamente competitivo.”(Gentil, 1993:p.51).

Segundo Santos (2012) a metodologia para melhorar a usabilidade dos softwares é verificar os fóruns da discussão de softwares *Open Source*, as comunidades *Open Source* onde os desenvolvedores caçam ideias consideradas boas expressadas por utilizadores das tais comunidades e estudá-las e no caso delas serem aprovadas serão implementadas.

Para que isso aconteça os desenvolvedores devem passar por diferentes etapas, conforme mostra a tabela abaixo:

Software <i>Open Source</i>	
Fases do DCU	Práticas de Usabilidade
Especificar o contexto de utilização	Modelo de pesquisa de utilizador Fórum de utilizadores Questionário
Especificar requisitos	Registro de soluções de desenvolvedores Registro de <i>brainstorm</i> ¹⁵ de utilizadores Versão Online do sistema
Criar soluções de <i>designer</i>	<i>Designer</i> por blog Diretrizes de interface com humanos Envolvimento de especialistas em <i>designer</i> de interface <i>Designer</i> cedo e frequente
Avaliar <i>designer</i>	Lançamento cedo para feedback dos utilizadores Reuniões de usabilidade Relatos de erros Testes de remotos

Tabela 2: Principais práticas de usabilidade utilizadas no desenvolvimento de software Open Source (Santos, 2012).

¹⁵ “Tempestade de ideias” é técnica de dinâmica de grupo, é uma atividade desenvolvida para explorar a potencialidade criativa de um indivíduo ou de um grupo.

Analisando a tabela 2, pode-se reparar que tudo inicia com os utilizadores, o que não é mau, visto que estes é que utilizam o software futuramente, mas recentemente isto não é feito apenas no desenvolvimento de softwares *Open Sources*.

4.5 Segurança

Segurança do software significa estabilidade e a confiabilidade em sua operação, qualidade, implementação e correção de falhas.

De acordo com Farias (2011) a estabilidade e a segurança são vantagens do software *Open Source* frente ao software proprietário.

Segundo Iwasaki (2008) *apud* Farias (2011) sua maior confiabilidade em relação ao software *Open Source* está no seu processo de desenvolvimento, que é coletivo. No entanto, há outros autores que reforçam a ideia de que o software *Open Source* possui uma boa segurança, que segundo Steward (2011), software *Open Source* permite a terceiros analisarem o código fonte com a finalidade de providenciar mais segurança para os utilizadores. Para além de Steward (2011) há autores como Sabino e Kon (2009) que afirma “ (...) dados olhos suficientes, todos os bugs são superficiais”. Isso significa que, com o maior número de utilizadores que têm acesso ao programa e até ao código-fonte, o software é testado melhor e os problemas existentes no código são encontrados mais rapidamente”.

Saltis (2009) afirma que o que põe as pessoas a verem os softwares proprietários como sendo mais seguros, é o facto de este ser desenvolvido num ambiente controlado, uma equipa concentrada.

Contudo, pode-se verificar o quadro abaixo o índice de vulnerabilidade dos softwares *Open Source* em relação aos softwares *Closed Sources* do mesmo tipo:

Application type	Product	#vuln	MTBVD [days]
Browser	Internet Explorer 7	74	13.29
	Firefox 2	167	5.16
Email client	MS Outlook Express 6	23	120.73
	Thunderbird 1	110	13.79
Web server	IIS 5	83	40.90
	Apache2	80	40.63
Office	MS Office 2003	99	19.22
	OpenOffice 2	19	63.16
Operating system	Windows 2000	385	9.35
	Windows XP	297	8.97
	MAC OSX	300	4.64
	Red Hat Enterprise Linux 4 ¹⁾	54 + 284 ²⁾ = 338	4.32
	Debian 3.1 ¹⁾	22 + 244 ²⁾ = 266	5.02
Database Management System	mySQL 5	33	46.00
	PostgreSQL 8	25	58.96
	Oracle 10g	63	29.72
	DB2 v8	13	136.38

Tabela 3: Número de vulnerabilidade de SOS e SP adaptado de Schryen (2009).

Reparando na tabela 3, na categoria *browser*, que o Mozilla Firefox sendo *Open Source* teve maior número de vulnerabilidade de que a Internet Explorer que é um software proprietário, mas já na categoria Office o MS Office teve maior número de vulnerabilidades frente a Open Office que é *Open Source*. O software *Open Source* destaca-se na categoria *Server* onde o servidor Apache como sendo um servidor *Open Source* teve menor número de vulnerabilidade de que IIS 5 da Microsoft que é um software *Open Source*.

4.6 Desenvolvimento

O desenvolvimento de software *Open Source* é um fenómeno recente, enquanto que o desenvolvimento de software de código fechado tradicional está aqui desde o início do desenvolvimento de software (Potdar & Chang, 2004:p.106), por outras palavras ainda há uma certa falta de documentação e metodologia de desenvolvimento *Open Source*.

Segundo Weerawarana *et al.*, (2004) participar em projetos *Open Source* permite que os desenvolvedores têm um impacto muito mais profundo e fundamental sobre a plataforma de computação do futuro do que através da construção de software de nível superior. O sucesso (ou fracasso) de um projeto *Open Source* é quase sempre caracterizado pela qualidade da comunidade de desenvolvedores em torno desse projeto (Weerawarana et al., 2004).

Ao falar de inovação tal como é apresentado na página 41, vê-se que o *Open Source* permite que pessoas ou empresas desenvolvam os softwares à partir do ponto onde outros desenvolvedores pararam. Raymond (1999:p.4) reforça essa ideia afirmando que “Linus Torvalds, por exemplo, não chegou a tentar escrever Linux a partir do zero. Em vez disso, ele começou com a reutilização de código e ideias do Minix, um sistema operativo Unix-like minúsculo para clones do PC.” Dessa ideia surgiu a famosa frase que diz que “Bons programadores sabem o que escrever. Os melhores sabem o que reutilizar”.

Dá uma certa impressão de que a forma de desenvolvimento do software *Open Source* resulta em melhor qualidade do software do que a forma proprietário, visto que os softwares *Open Sources* serão mais bem testados.

Para Raymond (1999) há dois modelos de desenvolvimento do software, o catedral e o bazar onde os SOS(software *Open Source*) adotam o modelo bazar, e o SP adota o modelo catedral.

A figura em baixo mostra as principais diferenças entre o modelo bazar e o modelo catedral, visto que cada modelo adapta a um tipo de software.

Modelo Catedral	Modelo Bazar
Trabalho controlado por um grupo de projetistas	Não há hierarquia entre os participantes
Estabelecimento de metodologias, tarefas e prazos	Projeto informalmente organizados ao redor de uma proposta inicial
Demora em lançamento de atualizações	Participação voluntaria de interessados
Dificuldade para atingir a massa critica	Atualizações constantes

Dificuldade para atingir qualidade esperada	Massa crítica atingida rapidamente
--	---

Tabela 4: Comparação entre os modelos Catedral e Bazar (Farias, 2011).

Ao analisar a tabela 4, pode-se ver que o modelo bazar providencia mais liberdade aos desenvolvedores, visto que não há hierarquia entre os participantes. No que diz respeito a atualizações, o modelo bazar disponibiliza mais atualizações do que o modelo catedral. No modelo catedral há grandes dificuldades em atingir a qualidade esperada, enquanto que no modelo bazar isto não acontece.

Em baixo será referido as principais vantagens e desvantagens que um software *Open Source* possui sobre o *Closed Source*.

4.7 Vantagens

Eis aqui algumas das vantagens do software *Open Source*, de acordo com Kon (2001):

- Baixo custo de aquisição ou nenhum em maioria dos casos;
- Software aberto promove a colaboração, isto é, troca de programas, informações e ideias entre a comunidade de usuários de um mesmo pacote ou Sistema;
- Possibilidade de utilizar recursos humanos altamente especializados em projetos que não teriam possibilidades comerciais a curto prazo;
- O software é testado por parte da comunidade de utilizadores que está mais interessada naquele sistema específico e, em geral, possui um conhecimento altamente especializado naquele campo;
- Relatório de erro mais completo, visto que o código fonte encontra-se disponível;

De acordo com Reijswoud e Jager (2008) os softwares *Open Source* também têm as seguintes vantagens:

- Redução de custos e menor dependência de tecnologia importada e habilidades;
- Software acessível para indivíduos, empresas e governo;
- O acesso universal através de distribuição do software em massa, sem implicações de licenciamento caros;

- O acesso aos dados do governo, sem barreira de formatos de software e dados proprietários;
- Capacidade de personalizar o software para línguas e culturas locais;
- *Patches* ou atualizações se tornam disponíveis mais rapidamente, o que limita as falhas e riscos de segurança;

4.8 Desvantagens

Vimos acima as principais vantagens que os softwares *Open Source* possui, algumas dela em relação ao software proprietário, ou melhor *Closed Source*.

De acordo com Farias (2011) os softwares *Open Source* apresenta algumas desvantagens:

- Interface de usuário não é uniforme nos aplicativos;
- Instalação e configuração pode ser difícil e mão-de-obra escassa e/ou custosa para desenvolvimento e/ou suporte, segundo Hexsel (2002) *apud* Farias (2011);

De acordo com Soares (2003) os softwares *Open Sources* em geral apresentam algumas desvantagens em relação ao software proprietário:

- Falta de uma instituição que responsabilize pelos softwares *Open Source*;
- Alguns dos fatores de extrema importância só são considerados no fim, como no caso da usabilidade;
- Pouca documentação sobre alguns softwares *Open Source*;

Segundo Weerawarana et al. (2004) uma das principais desvantagens que o *Open Source* tem é:

- Pelo facto de existir muitas tecnologias que desempenham a mesma função e poucas informações credíveis que nos diga qual é melhor para usarmos na produção.

Reijswoud e Jager (2008) identifica ainda outras desvantagens:

- Documentação: A documentação que acompanha software *Open Source* é muitas vezes idiossincrática e, às vezes inexistente. Desenvolvedores de software *Open Source* estão motivados para os aspetos técnicos da aplicação do que para a usabilidade.

- Ajustamento entre Hardware e software: software *Open Source* muitas vezes fica para trás sobre novo hardware. Isto é causado pelo facto de os fabricantes de hardware não conseguirem liberar as especificações de hardware em tempo para a comunidade de software *Open Source*.

5 Software Open Source nos países em desenvolvimento como Cabo Verde

“Nos países em desenvolvimento é elevada a taxa de utilização ilegal de software, isto é, há elevada utilização de software proprietário sem o devido licenciamento” (I. B. O. da Rosa, 2010). Weerawarana et al. (2004) afirma ainda que nos países em desenvolvimento normalmente os softwares proprietários é que dominam o mercado com ou sem licença, isto porque os softwares *Open Sources* não possuem recursos suficientes para fazer uma forte campanha de marketing necessária para permitir uma escolha entre os softwares *Open Source* e softwares *Closed Source*. Por isso, em muitos países em desenvolvimento muitos utilizadores não importam muito se um software seja *Open Source* ou *Closed Source*, visto que eles não pagam por softwares pago; essa visão mudaria se as pessoas pagassem por softwares proprietários. “Em Cabo Verde, por exemplo, o preço do pacote do Windows XP + Office 2007 é US\$ 760, quando o salário mínimo nacional em discussão é de cerca de US\$ 187!” (I. B. da Rosa & Mendes, 2010). No entanto, os valores apresentados acima convertidos em escudos, onde cada um dólar americano equivale a aproximadamente 83 escudos cabo-verdianos, o preço do Windows XP + Office 2007 é aproximadamente 63 mil escudos, o que ultrapassa o salário mínimo que é 11 mil escudos.

Isto quer dizer que se for para pagar todos os softwares que utilizamos no dia-dia, teríamos grandes dificuldades, visto que a maioria de nos não possui recursos financeiros suficiente para comprar esses softwares proprietários.

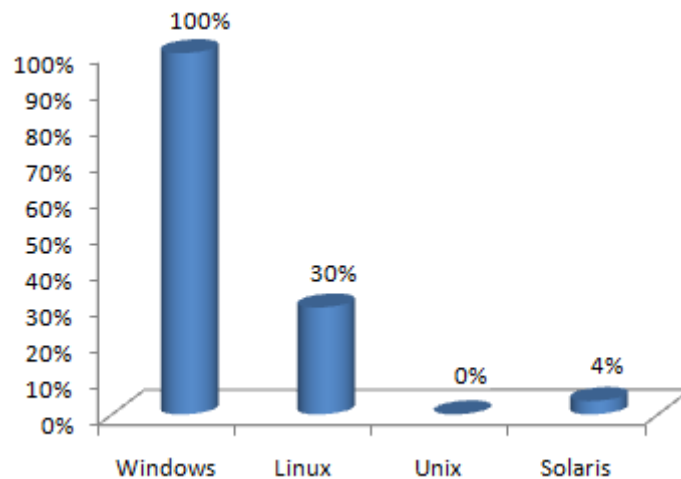


Figura 6: A utilização do sistema operativo de desktop (I. B. da Rosa & Mendes, 2010).

Pode-se ver na figura 6, o índice da utilização de sistemas operativos em Cabo Verde, que é um país em desenvolvimento, onde a utilização de sistemas proprietários (Windows) é maioritário com 100% de utilização e em segundo lugar o Linux com 30% de utilização, que é um software *Open Source*. De acordo com (I. B. da Rosa & Mendes, 2010) as quatro principais razões que determinam a utilização desses sistemas operativos são: a disponibilidade de aplicativos (63% dos entrevistados), a facilidade de utilização (52%), software pré-instalado (52%) e disponibilidade de suporte técnico (44%).

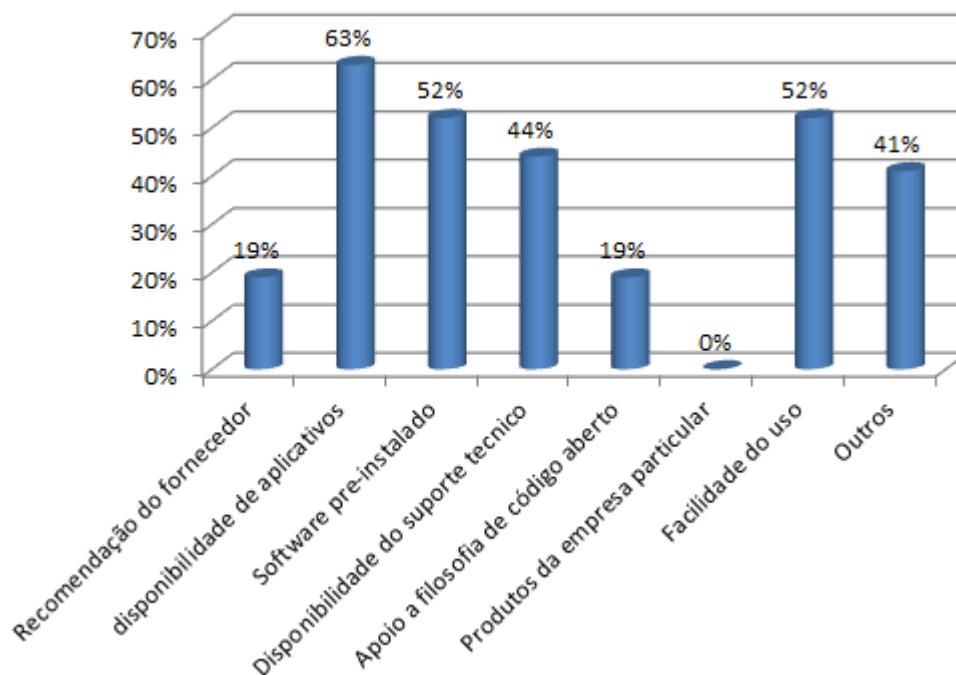


Figura 7: Razões da utilização de sistemas operativo de desktop (I. B. da Rosa & Mendes, 2010).

Na figura 7, pode-se ver que as pessoas não dão muita importância se o software a utilizar é *Open Source* ou não, mas dão mais importância a critérios como disponibilidade do software, facilidade da utilização, pré-instalação do software, disponibilidade de suporte técnico.

5.1 Oportunidade para países em desenvolvimento

Nos países desenvolvidos a tendência é de serem mais desenvolvidos em termos de tecnologias, visto que possuem mais recursos, o que lhes possibilitam pagar uma mão-de-obra mais qualificada, e até importá-la no caso de não haver. Porém, os softwares *Open Source* fazem com que países em desenvolvimento desenvolvam mais rapidamente em termos de tecnologias, dado que podem pegar num software *Open Source* e estudá-lo. Agora vamos apresentar as principais oportunidades que os softwares *Open Sources* oferecem para países em desenvolvimento:

- Os softwares *Open Sources* oferecem uma abordagem estratégica global para ser abordada pelos países em desenvolvimento na busca de criação de valores na economia através de TI (Weerawarana et al., 2004);
- Segundo Kon (2001) os softwares *Open Source* ajudam a diminuir a enorme distância entre os países desenvolvidos e países em desenvolvimento, pode-se tomar como exemplo Estados Unidos e Cabo Verde;
- (...) Oportunidades sem precedentes para a indústria de software de país em desenvolvimento para desenvolver a capacidade e posicionar-se para aproveitar as oportunidades globais (Weerawarana et al., 2004);
- Ao optar por utilização de softwares *Open Source* nos países em desenvolvimento no sector público, reduz a dependência que esses países têm em relação aos softwares proprietários e aborda a aplicação dos direitos de propriedade intelectual (Webe;2003) *apud* (Weerawarana et al., 2004);
- A utilização dos softwares *Open Source* permite que o suporte e manutenção sejam contratados livremente por parte das empresas dos países em desenvolvimento o que permite que estes contratem livremente numa gama de muitos fornecedores de qualidade e a um custo mais acessível para serviços tipo instalação, manutenção e suporte, sabendo que tudo isto visa minimizar os gastos desses países (Weerawarana et al., 2004:p.28). Reforçando está ideia o mesmo autor (*idem*)“a manutenção é ainda

replicável sem incorrer em custos grandes, uma vez que modificações no código fonte também são livres”;

- Maior ou novas oportunidades de negócios no sector de TI para empresas privadas;
- Redução dos custos de TI na economia, tanto para o governo como para empresas privadas;
- Melhoria da eficácia e eficiência na governação eletrónica (Weerawarana et al., 2004);

6 Considerações finais

No presente Capítulo, vimos o que é um software *Open Source*, o seu surgimento e que o difere dos softwares proprietários. Depois de falar do conceito da *Open Source*, *Closed Source*, historial de ambos, também fazemos uma pequena comparação entre os dois tipos de softwares levando em conta os seguintes critérios: custo, serviço/suporte, inovação, usabilidade, segurança e desenvolvimento. Essa tal comparação foi feita com o objetivo de expor as vantagens e desvantagens que cada tipo de software tem sobre o outro.

Elaborámos também as principais vantagens e desvantagens que o software *Open Source* apresenta de uma forma generalizada.

Como vimos, o custo de obtenção e de manutenção é uma das preocupações dos clientes em relação aos softwares; fez-se necessário falar dos softwares *Open Source* nos países em desenvolvimento como Cabo Verde, onde sabemos que o poder de compra é inferior aos países desenvolvidos, assim como o desenvolvimento no que se refere aos fatores tecnológicos. Contudo, vimos que os softwares *Open Source* oferecem algumas oportunidades especiais aos países menos favorecidos a nível económico e a nível tecnológico.

Capítulo 2: Tecnologias Open Source

Vivemos numa época em que quase tudo gira à volta das tecnologias, visto que utilizamos tecnologias para fazer quase tudo, para trabalhamos, para estudamos, para divertimos, ou melhor, em tudo quanto é lado e no seio dessas tecnologias há uma forte influência de tecnologias *Open Source*. Na maioria das tecnologias proprietárias há uma alternativa *Open Source* de boa qualidade nem sempre muito utilizada. Neste Capítulo vamos falar um pouco das tecnologias *Open Source* na categoria *browsers*, servidores, sistemas operativos, base de dados e editores onde vamos estudar quando é que surgiram, citar as suas características. Nas estatísticas abaixo pode-se reparar que há uma forte liderança em maioria das categorias citadas. Weerawarana et al. (2004) afirma que há três grandes segmentos de softwares e estes encontram-se fortemente dominados por duas empresas proprietárias quais sejam Microsoft e IBM, e os tais segmentos são:

- Sistemas operativo de servidores e aplicações;
- Sistemas operativo de desktop / cliente e aplicativos;
- Soluções empresariais.

Mas, neste trabalho, para que isto fique mais detalhado foi elaborado uma classificação por tipo de softwares conforme é citado abaixo:

- *Web browser*;

- IDEs;
- Servidor *Web*;
- Editores;
- Base de dados;
- Sistemas operativos;

1 Web browsers

Hoje em dia há uma grande guerra entre as empresas que desenvolvem *browsers*, como por exemplo Microsoft, Apple, Google entre outros, e nessa correria encontra-se também *browsers Open Sources* como no caso da famosa Mozilla Firefox. A escolha recaiu sobre o referido browser por ser considerado um dos melhores da atualidade (ver página 55 figura 8).

1.1 Mozilla Firefox

Segundo Landim (2011) Mozilla Firefox é um *browser* (navegador) *Open Source* mantido pela Mozilla Foundation, onde a primeira versão foi lançado em 9 de Novembro de 2004. A Mozilla surgiu à partir dos códigos partilhados na comunidade *Open Source* da Netscap Communicator (Negus, 2010). Yeow (2005) acrescenta ainda que, depois do surgimento da Mozilla surgiu a Mozilla Firefox que é o *browser* que conhecemos hoje.

Características:

As características a seguir são apontadas por (Yeow, 2005):

- Tabulação de páginas, o que permite a um utilizador abrir múltiplas páginas numa só janela;
- Sistema nativo de bloqueio de pop-up;
- Sistema de bloqueio de Javascript, o que permite que utilizadores bloqueiem algumas chamadas de Javascript;

- Gerenciador de *download* nativo, o que permite a realização de *download* simples e rápido;
- *Plugins*, o Mozilla Firefox permite a instalação de *plugins* e aumentar a funcionalidade do *browser*, permitindo que este seja capaz de abrir ficheiros do tipo PDF, Flash, Java applets, entre outros;

Podemos conferir a popularidade do Mozilla Firefox na figura em baixo:

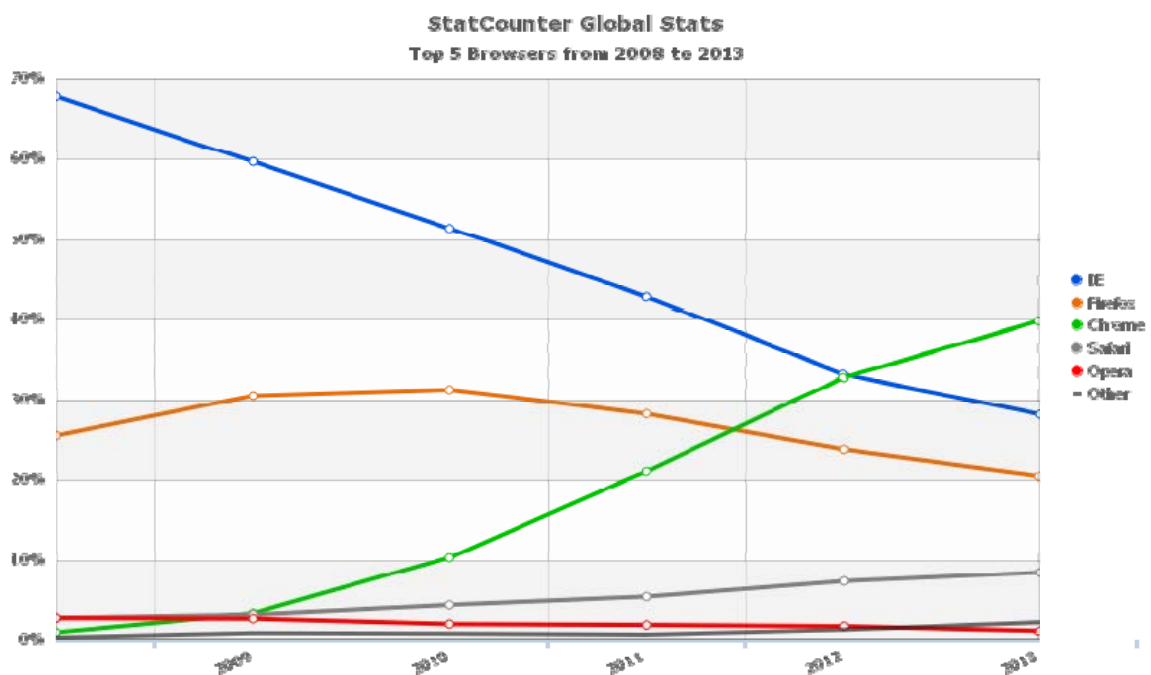


Figura 8: Top 5 browser de 2008 à 2013 (Stats, 2013a)

De acordo com a figura 8 Mozilla Firefox é o *browser Open Source* mais utilizado tendo uma parcela de 22% dos utilizadores a nível mundial, no top 5 *browser* mais utilizado em geral ou seja juntamente com outros *browser* não *Open Source*, ele fica atrás do Internet Explorer e Google Chrome.

Na figura 8 também pode-se ver bem claro o progresso fantástico do Google Chrome que não é *Open Source* e que em 2011 colocou o Mozilla Firefox na terceira posição que antes estava na segunda posição.

2 IDEs

IDE¹⁶, do inglês *Integrated Development Environment*, que ao traduzir para português fica “ambiente integrado de desenvolvimento”, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar o processo de desenvolvimento de softwares.

2.1 Eclipse

Eclipse¹⁷ é um IDE desenvolvido em Java, seguindo o modelo *Open Source* de desenvolvimento de softwares. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do software e doou-o como software *Open Source* para a comunidade.

Segundo (Foundation, 2013), a Fundação Eclipse foi criada em Janeiro de 2004 como uma organização independente, corporação sem fins lucrativos para atuar como administrador da comunidade Eclipse. Mas Faria et al. (2010) afirma que o desenvolvimento do eclipse iniciou-se no ano 2001, em suma a fundação apareceu 3 anos depois.

Características

- É multilinguagem;
- Permite a *refactoring* do código, que é uma forma organizada de reestruturar o código para ser melhorado;
- Eclipse utiliza a plataforma SWT;
- É multiplataforma ou seja pode ser instalado numa distribuição Linux, Windows e Mac OS (Faria et al., 2010).

¹⁶Disponível em [http://pt.wikipedia.org/wiki/Ambiente_de_Desenvolvimento_Integrado], consultado a 17 de Julho de 2013

¹⁷ Disponível em [[http://pt.wikipedia.org/wiki/Eclipse_\(Software\)](http://pt.wikipedia.org/wiki/Eclipse_(Software))], consultado a 08 de agosto de 2013

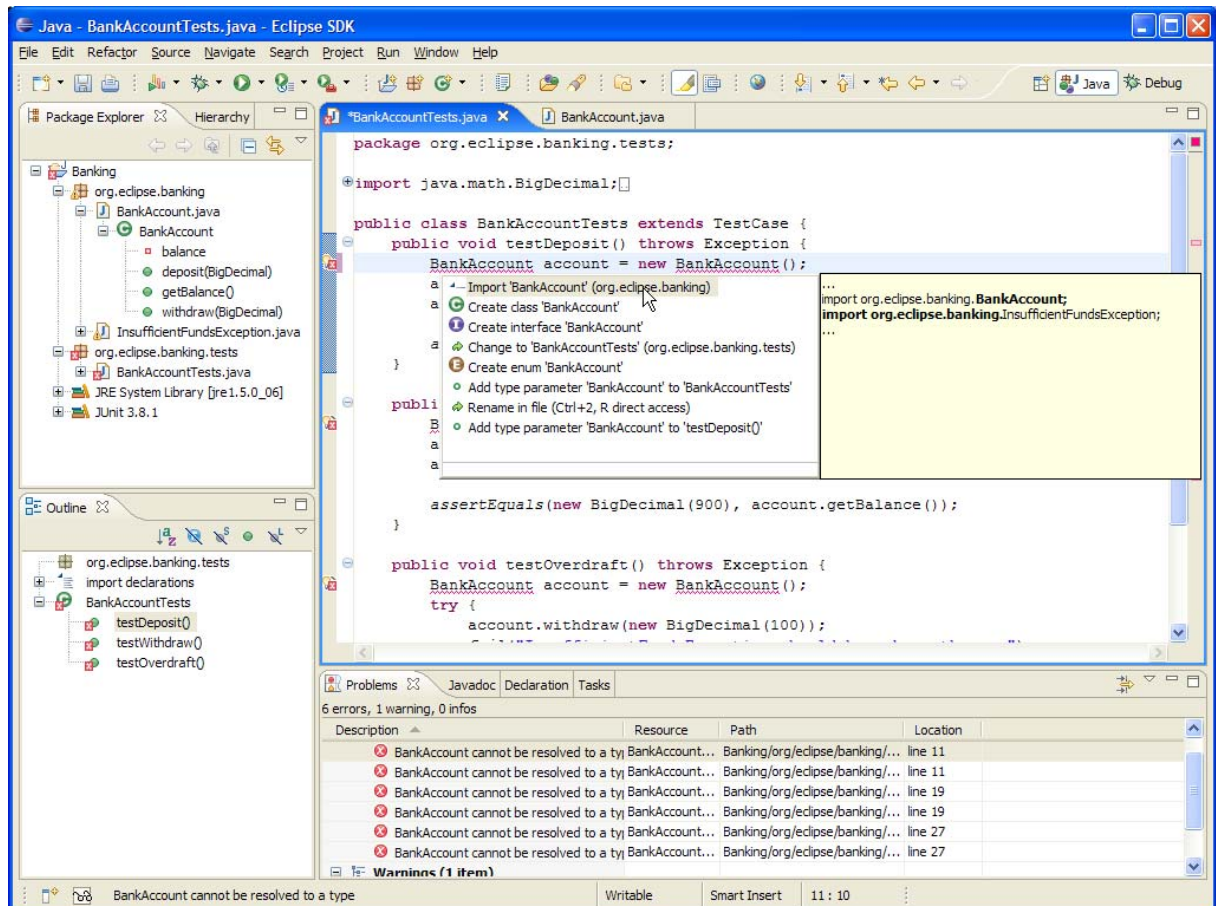


Figura 9: Interface do Eclipse (ApacheORG, 2013)

2.2 Netbeans

NetBeans é um IDE *Open Source* gratuito para fins lucrativo e fins não lucrativo desenvolvido por um grupo de estudantes da República Tcheca em 1996. O NetBeans oferece aos desenvolvedores ferramentas indispensáveis para criar aplicativos profissionais de desktop, empresariais, *Web* e móveis multiplataformas. Além de multiplataformas ela também tem editores e ferramentas para linguagens além Java como XML, HTML, PHP, Groovy, Javadoc, JavaScript, C e JSP (NetBeans, 2013).

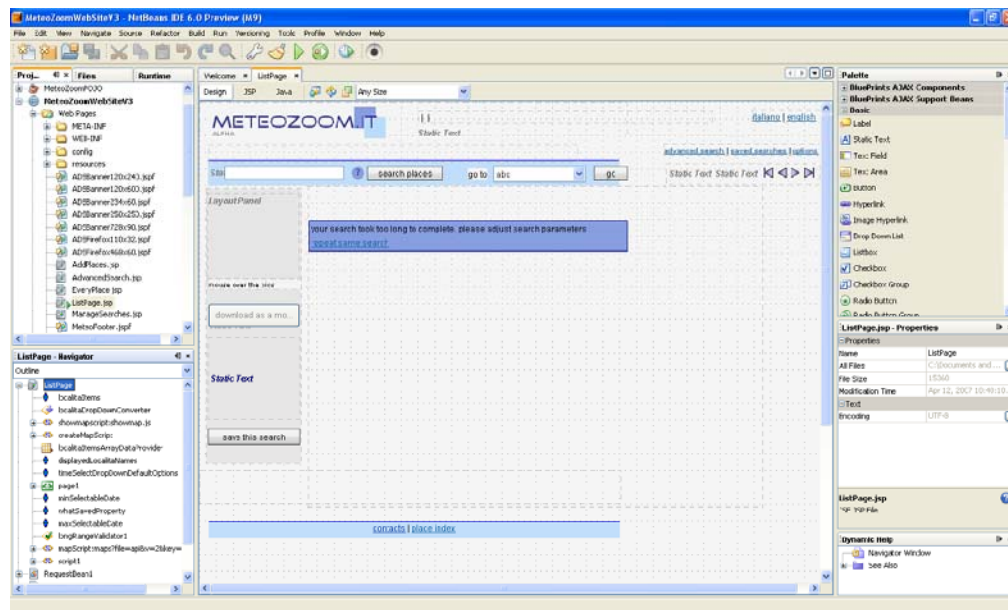


Figura 10: Interface do Netbeans 6.0 (NetBeansORG, 2011)

Característica

De acordo com Böck (2009) as principais características da plataforma Netbeans são:

- Grátis;
- Interface do IDE personalizável de modo a permitir que os utilizadores personalizem o programa de acordo com as suas necessidades;
- Dispõe de um sistema de ajuda muito eficiente que exhibe os tópicos de ajuda para os utilizadores finais;
- Permite *refactoring* de código fonte.

3 Servidor Web

“Um servidor *Web* pode significar duas coisas - um computador em que um *site* está hospedado e um programa que roda em um computador como esse. Assim, o termo servidor Web se refere a hardware e software” (Webdevelopersnotes, 2013). Existe muitos servidores *Open Source*, mas nós na realização desse trabalho escolhemos o servidor Apache porque de acordo com o gráfico que mostra a sua popularidade (figura 4) é um dos servidores Web mais utilizado da atualidade.

3.1 Apache

Apache é o mais bem-sucedido servidor *Web* de código fonte aberto ou seja *Open Source* e gratuito. Segundo Apache Foundation (ApacheFoundation, 2013b) o servidor Apache foi desenvolvido em Fevereiro de 1995 por Rob McCool no National Center for Supercomputing Applications, da Universidade de Illinois.

Características

- Suporte multi-protocolo;
- Suporte aperfeiçoado para plataformas não-Unix;
- Nova API Apache;
- Suporte IPv6;
- Configuração simplificada;
- Respostas de erro multilíngues;
- Suporte nativo ao Unicode¹⁸ do Windows NT;
- Biblioteca de expressões regulares atualizada (ApacheFoundation, 2013a).

4 Editores

Programa responsável por apresentar texto numa página ou no visor, a medida que se vai escrevendo e se vão fazendo alterações necessárias (Costa e Melo, 1999). No caso dessa afirmação esta a referir a um editor de texto, um editor de imagem faz o mesmo que um editor de texto só que é com imagens. No entanto para realização desse trabalho escolhemos EMACS visto que este é um software da GNU e GIMP como sendo o principal rival do Photoshop segundo o site (AlternativeTo, 2014).

¹⁸ Unicode é um padrão que permite aos computadores representar e manipular, de forma consistente, texto de qualquer sistema de escrita existente.

4.1 EMACS

GNU Emacs é um editor de texto personalizável e extensível. Na sua essência é um interpretador para Emacs Lisp, um dialeto da linguagem de programação Lisp com extensões para apoiar a edição de texto (GNU, 2013). Richar Stallman (1981) explica que extensível significa que utilizadores podem adicionar novos comandos de edição ou mudar os velhos comandos para atender as suas necessidades.

Richar Stallman (*idem*) afirma ainda que a capacidade de estender faz com que o EMACS seja mais flexível que os outros editores de texto.

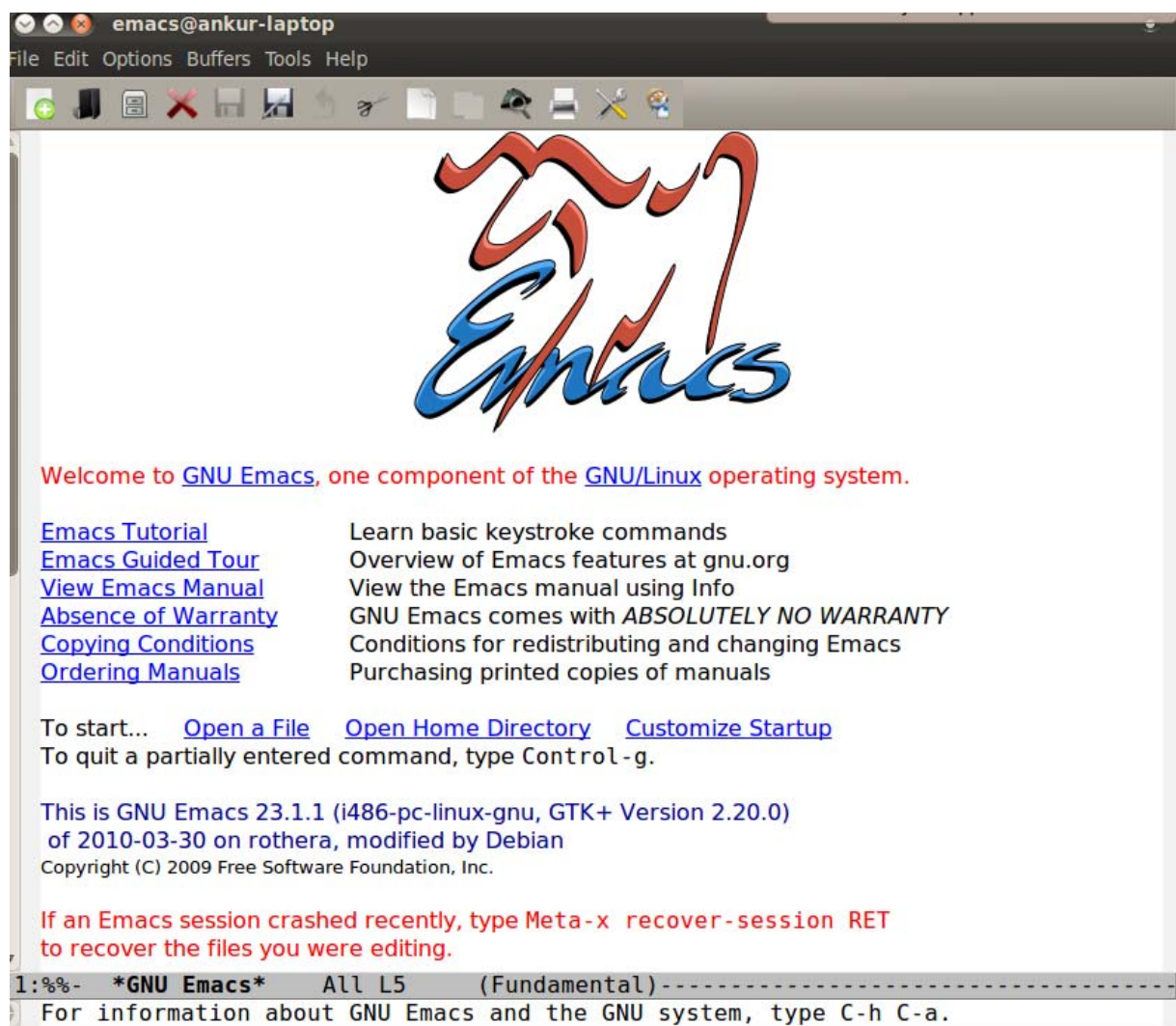


Figura 11: Interface do EMAC 23.1.1 (GNU, 2013)

Características

- Um sistema de embalagem e de interface para baixar e instalar extensões;

- Suporte para exibir e editar texto bidirecional, incluindo scripts da direita para a esquerda, como árabe e hebraico;
- Suporte para delimitação lexical em Emacs Lisp;
- Melhorias no sistema de temas personalizados;
- Sistema unificado de conclusão e melhor em muitos modos e pacotes (GNU, 2013).

4.2 GIMP

GIMP é um acrónimo para GNU “*Image Manipulation Program*” que ao traduzir para português fica programa de manipulação de imagem. É um programa livremente distribuído para tarefas como o retoque de fotos, composição de imagem e criação de imagem (Gimp, 2013). A mesma fonte afirma que GIMP foi desenvolvido inicialmente em 1995 por Peter Mattis e Spencer Kimball. A figura abaixo ilustra a interface do GIMP.



Figura 12: Interface do GIMP 2.8.10 (fonte: <http://www.gimp.org/screenshots/>)

Características

- Conjunto completo de ferramentas de pintura, incluindo pincéis, um lápis, um aerógrafo, clonagem, etc.
- Gerenciamento de memória baseado em Azulejo para o tamanho da imagem é limitada apenas pelo espaço disponível em disco;
- Amostragem sub-pixel para todas as ferramentas de pintura de alta qualidade *anti-aliasing*;
- Suporte ao canal *alpha* completo;
- Camadas e canais;
- Possui uma base de dados processuais para chamar funções do GIMP internos de programas externos, como o Script-Fu;
- Recursos de script avançada;
- Múltiplo desfazer / refazer (limitada apenas pelo espaço em disco);
- Ferramentas de transformação, incluindo rotação, escala, cisalhamento e *flip*;
- Os formatos de arquivos suportados incluem GIF, JPEG, PNG, XPM, TIFF, TGA, MPEG, PS, PDF, PCX, BMP e muitos outros;
- Carregar, exibir, converter e salvar a muitos formatos de arquivo;
- Ferramentas de seleção incluindo retângulo, elipse, livre, distorcido, bezier e inteligente;
- *Plugins* que permitem a fácil adição de novos formatos e novos filtros de efeitos (Kylander e Kylander, 1999).

Muitas dessas características podem se verificar que estão presente no Photoshop que é um software quase semelhante ao GIMP só que é proprietária pertencente a Adobe e muito mais utilizado que o GIMP. Pode-se ver uma comparação feita por (Kylander e Kylander, 1999:p.3):

Comparado com Photoshop, GIMP tem tudo (e ainda mais se você não comprar *plugins* de terceiros para Photoshop). A maior parte dos recursos do GIMP é mais flexível e poderosa, uma vez que você começa a conhecê-los. A grande coisa é que GIMP suporta o formato de arquivo PSD e arquivos de fábrica de filtro, assim você pode facilmente mudar de Photoshop para o GIMP. Simplesmente, é um pedaço de um programa e ele vem carregado com um saco de *plugins*. Então, ir buscá-lo!

Você não vai se decepcionar, e, bem, não é errado, que é gratuito.

No caso do parágrafo acima o autor não só estava equiparando o GIMP com Photoshop, mas sim fazendo um apelo para que utilizadores abandonem o Photoshop para utilizar GIMP, mostrando a questão dos custos, visto que usuário do Photoshop além de comprarem a licença, comprem também os *plugins* e isso no GIMP é gratuito, ele também diz que GIMP suporta o formato PSD e Photoshop não e ele também diz que os recursos de GIMP é mais flexível e poderoso e por fim diz que não vamos arrepender ao fazer a mudança de Photoshop para GIMP.

5 Base de dados

De acordo com Colares (2007), base de dados é uma coleção de dados que mantém relações entre si e está armazenada em um dispositivo que permite armazenar, modificar e recuperar dados.

Wikipédia (2013a) afirma que uma base de dados é operada pelos Sistemas Gerenciadores de Bancos de Dados (SGBD), que surgiram na década de 70.

Pode-se assim afirmar que uma base de dados é um conjunto ou coleções de dados que mantem um relação entre si e possui uma SGBD. Para realização deste trabalho escolhemos os SGBDs *Open Source* mais populares segundo a tabela 5 que ilustra o ranking mundial de SGBDs.

5.1 Mysql

Mysql é um SGBD relacional que utiliza a linguagem padrão SQL (*Structured Query Language*) e é largamente utilizado em aplicações para a Internet (Niederauer, 2006).

Fernandes (2008), acrescenta ainda que o Mysql é um RDBMS (*Relational Database Management System* – que ao traduzir para português fica Sistema de Gestão de Base de Dados Relacional), que está disponível para diferentes sistemas operativos Windows, Linux e outros (...). De acordo com (Arjen Lentz, DuBois, & Hinz, 2006) o Mysql foi concebida originalmente por David Axmark, Allan Larsson e Michael “Monty” Widenius na Suécia.

Mysql é sem duvida uma das referencia do software *Open Source* tanto em termos de utilização, quanto em termos de dimensão do software (ver tabela 5).

Características

- Número ilimitado de utilização por utilizadores simultâneos;
- Capacidade de manipulação de tabelas com mais de 50.000.000 de registos;
- Alta velocidade de execução de comandos;
- Fácil e eficiente controlo de privilégios de usuários (Niederauer, 2006).

Fernandes (2008:p.56) acrescenta algumas características de Mysql:

- Facilidade na instalação, visto que este não exige muita configuração;

- Não precisa de muito recurso para correr. A velocidade com que pode disponibilizar a informação faz dele o favorito de muitos profissionais da informática;
- Independência nas uniões: Pode-se fazer queries que junta tabelas de diferentes bases de dados;
- Tem uma API para maioria de linguagem de programação. Em especial pode-se escrever aplicações para aceder a bases de dados Mysql em C, C++, Eiffel, Java, Perl, PHP, Python e TcL;

Pode-se conferir a popularidade do Mysql numa lista de mais de uma centena de SGBD e este aparece em 2º lugar, o que já é alguma coisa:

Rank	Last Month	DBMS	Database Model	Score	Changes
1.		1. Oracle	Relational DBMS	1544.44	+16.80
2.	↑	3. MySQL	Relational DBMS	1324.83	+19.59
3.	↓	2. Microsoft SQL Server	Relational DBMS	1304.96	-17.02
4.		4. PostgreSQL	Relational DBMS	182.22	-0.16
5.		5. DB2	Relational DBMS	162.94	-5.49
6.	↑	7. MongoDB	Document store	155.99	+15.53
7.	↓	6. Microsoft Access	Relational DBMS	150.88	+9.24

Tabela 5: Ranking dos principais SGBD de agosto de 2013 (Ranking-DB-Engines, 2013).

De acordo a tabela 5, que mostra a classificações mundial de SGBDs, o Mysql subiu de terceiro para o segundo lugar ultrapassando Microsoft SQL *Server*, ficando atrás da SGBD proprietário Oracle. Isto prova nos que os softwares *Open Source* estão cada vez com mais popular.

5.2 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional (SGBDOR) baseado no POSTGRES, versão 4.21, desenvolvido na Universidade da Califórnia em Berkeley Computer Science Department (Group, 2013).

PostgreSQL antes era chamado de Postgre foi desenvolvido por um professor de ciência da computação chamado Michael Stonebraker em 1986 (*idem*).

Pode-se ver a sua popularidade na tabela 5, no meio de mais de 100 aplicativos de gestão de base de dados e este se encontra na 4ª posição, ficando atrás de Microsoft SQL *Server* que

ocupa a 3ª posição, MySQL que ocupa 2ª posição que também é *Open Source* e Oracle que não é *Open Source* porém que é líder dessa lista.

Características

Ele suporta grande parte do padrão SQL e oferece muitas características modernas:

- Gatilhos;
- Consultas complexas;
- Integridade transacional;
- Chaves estrangeira (Group, 2013).
- Ele também tem uma série de extensões e recursos avançados;
- Fornece uma interface que permite a criação de tipos de dados personalizados;
- Herança de tabelas, o que permite que *designers* de base de dados geral novas tabelas à partir de outras tabelas;
- PostGIS é um projeto que adiciona suporte para objetos geográficos no PostgreSQL, permitindo que ele seja utilizado como um banco de dados espacial para sistemas de informação geográfica (GIS) (“PostgreSQL: About,” 2014).

6 Sistemas operativos

De acordo com Jandl (2004), um sistema operativo é um conjunto de programas implementados como softwares ou *firmware*¹⁹ que tornam o hardware utilizável. Bacic (2003:p.9) afirma que um sistema operativo “é o principal programa do computador, sendo de sua responsabilidade gerenciar e monitorar os recursos de um computador. É ele que "diz" ao computador como utilizar os periféricos e rodar os programas”. Porém pode-se dizer perfeitamente que o sistema operativo é que dá vida aos computadores (Hardware), visto que um computador só é utilizado por uma pessoa normal se este se encontra com um sistema operativo instalado. Na categoria sistemas operativos escolhemos o Kernel Linux devido a sua popularidade (ver pagina 67 figura 14), único sistema operativo Open Source da lista.

¹⁹ **Firmware** é o conjunto de instruções operacionais programadas diretamente no hardware de um equipamento eletrônico (Wikipédia, 2013b).

6.1 Linux

Linux é um núcleo (kernel) do sistema operativo desenvolvido pelo programador finlandês de nome Linus Torvalds (Wikipédia, 2013c). Segundo Ceruzzi (2003) Linux é uma variante de Unix. KernelOrg (2013) afirma ainda que Linux é um clone de Unix, mas ele foi escrito de zero, com a ajuda vagamente de uma equipa de *hackers* espalhado no mundo inteiro.

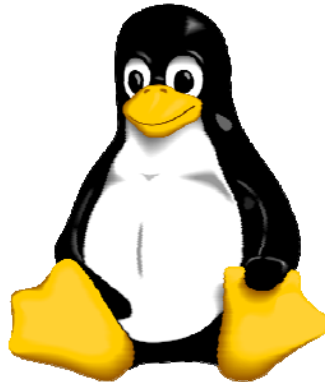


Figura 13:Tux, mascote do Linux (Wikimedia, 2013c).

Características de Linux

Essas são algumas características de Linux apresentada por (Garrels, 2008):

- É portátil para qualquer plataforma de hardware;
- Foi feito para continuar a correr;
- É seguro e versátil;
- É escalável;
- O sistema operativo Linux e a maioria das aplicações do Linux têm tempos de *debug* muito curtos;
- Existência de muitas distribuições diferentes, cada um com o seu propósito;
- Não é muito amigável e confuso para iniciantes;
- É um software *Open Source* confiável;

Podemos ver o índice da utilização do Linux entre 2008 a 2014 que ela não é muito utilizada (ver figura 14).

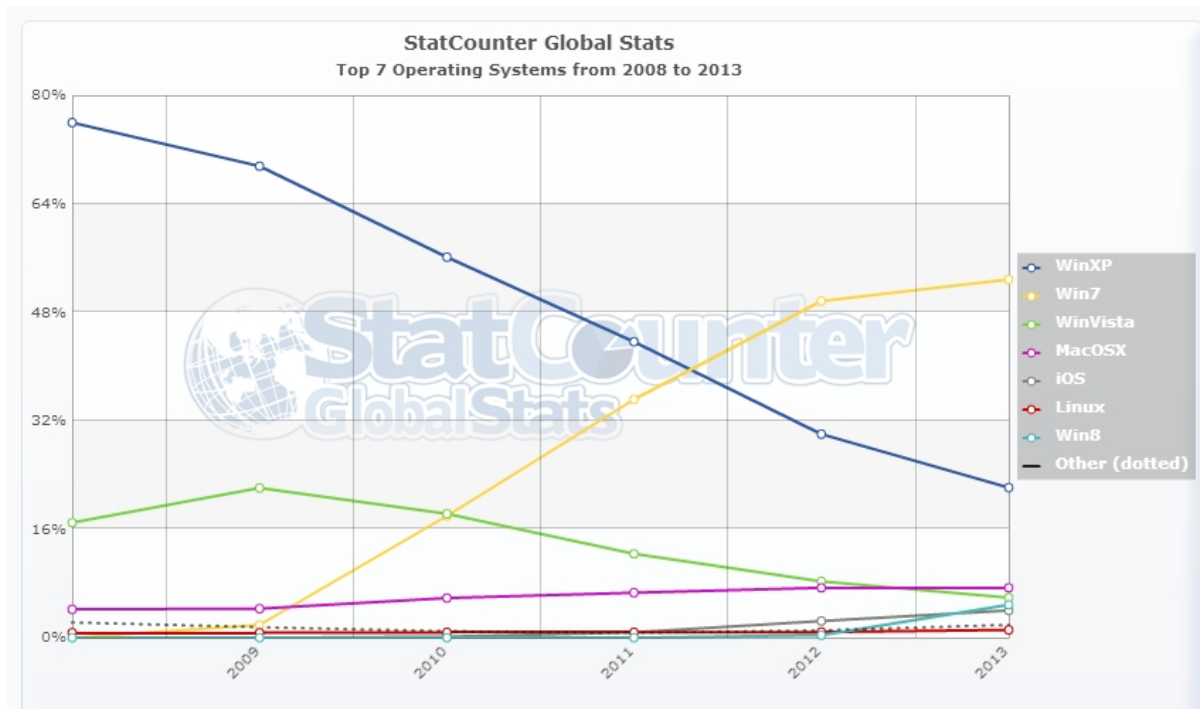


Figura 14: Top 7 sistemas operativos 2008 à 2013 (Stats, 2013b).

O gráfico da figura 14 indica-nos o índice da utilização dos sistemas operativos a nível mundial, onde se destacam os principais sistemas operativos atuais, e de acordo com esse gráfico o Linux como único sistema operativo *Open Source* da lista encontra-se na 7ª posição atrás de iOS que possui 2 por cento, no entanto o Linux possui um valor muito baixo em comparação com o Windows 7 que é o sistema operativo mais utilizado no momento ou com o Windows XP que é o segundo sistema operativo mais utilizado. Segundo (Weerawarana et al., 2004) a posição do Linux nesse caso se justifica no facto do Linux não possuir recursos suficientes para fazer uma forte campanha de marketing como no caso de Windows. Se repararmos no nosso quotidiano, pode-se ver que muitas pessoas fazem especulações em relação ao lançamento do Windows, as datas do lançamento, o que não acontece em grande escala com as desatribuições Linux.

6.2 História do Linux

“Todos os sistemas operativos modernos têm suas raízes em 1969, quando Dennis Ritchie e Ken Thompson desenvolveram a linguagem C e o sistema operativo Unix da AT & T Bell Labs.” (Cobbaut, 2013:p.3)

Então vamos fazer uma pequena viagem no tempo onde vamos estudar a origem de tudo isso que é Unix. Segundo Garrels (2008) em 1969 uma equipa de desenvolvedores do laboratório de Bell Labs desenvolveu um novo sistema operativo chamado Unix que era:

- Simples e elegante.
- Escrito na linguagem de programação C em vez de código de montagem.
- Capaz de reciclar código.

Porém nos anos oitenta, muitas empresas começaram a desenvolver seu próprio Unix: IBM criou AIX, Sun SunOS (mais tarde Solaris), HP HP-UX e cerca de uma dúzia de outras empresas fizeram o mesmo. O resultado foi uma confusão de dialetos Unix e uma dúzia de maneiras diferentes de fazer a mesma coisa. Cobbaut (2013) (...) ou seja tudo era Unix modificado, ou melhor dizer essas empresas não construíram sistemas operativos de raiz, e para acabar com essa confusão vem o Rich Stallman.

E aqui está a primeira raiz real do Linux, quando Richard Stallman teve como objetivo acabar com essa era de separação Unix e todos reinventar a roda, iniciando o projeto GNU (GNU não é Unix). Seu objetivo era fazer um sistema operativo para ser disponível gratuitamente para todos, e onde todos podem trabalhar em conjunto (como nos anos setenta). Muitas das ferramentas de linha de comando que utilizamos hoje em Linux ou Solaris são ferramentas da GNU; Cobbaut(2013:p.3).

Segundo Rob (2013) o motivo pela qual levou o Linus a criar Linux, foi quando ele ainda estava na Universidade de Helsinki, estavam a utilizar versão do sistema operativo UNIX chamado Minix, ele e mais um colega enviaram pedido de modificações e melhoria ao criador do Minix Andrew Tanenbaum, mas este recusou, foi então que Linus resolveu criar o seu próprio sistema operativo.

Na década de 90 Linus Torvalds iniciou um novo projeto da construção de um novo núcleo compatível com o POSIX, e disse que vai disponibilizar o código fonte (Cobbaut, 2013).

De acordo com Reijswoud e Jager (2008) o projeto só foi lançado em 1991, recebendo o nome de Linux. Pode-se dizer que o mundo só teve conhecimento do Linux a partir de 1991.

No início o Linux era utilizado apenas por profissionais da computação, *geeks* e *hackers* apesar de a situação só inverteu uma década depois onde muitos dão colaboração para o melhoramento do Linux fazendo dele melhor e mais utilizado (McCallister, 2008).

O mesmo autor (*idem*) afirma que o Linux é a principal alternativa de sistema operativo grátis e *Open Source*.

6.3 Distribuições Linux

Segundo Cobbaut (2013) uma distribuição Linux é qualquer sistema operativo (*Open Source*), que utiliza o Kernel do Linux. “As distribuições Linux são compiladas por indivíduos, grupos interligados em longa distância e varias organizações profissionais. Geralmente incluem: softwares adicionais, aplicativos, um instalador do sistema com uma tela fácil manusear e componentes simplificados para gerenciamento de atualização” (Iwata, 2009:18). Normalmente pode-se ver nas alíneas em baixo que nem todas as distribuições têm uma empresa por detrás.

Agora vamos conhecer as principais distribuição do Linux apontadas por (Negus, 2010) :

- **Fedora** - ²⁰É um sistema operativo que tem por base o kernel Linux. Foi criada em 2003 pela Red Hat. Atualmente mantida pelo projeto Fedora. As primeiras versões do sistema desenvolvidas foram Fedora Core 1-4, que foi lançado entre 2003 e 2005, o Fedora Core 5-6 lançado em 2006, o Fedora 7 em 2007 não foram suportadas pelo projeto Fedora. As versões posteriores como Fedora 8 lançada em 2007, o Fedora 9 lançada na Primavera de 2008 e Fedora 10 lançada no final de 2008, são todas suportadas oficialmente pelo projeto Fedora.
- **Red Hat Enterprise Linux** -²¹ Pode ser considerada uma versão corporativa do antigo Red Hat Linux, criada após a empresa se desvincular do mercado privado, iniciando o projeto Fedora, e se concentrar no mercado corporativo.
- **Debian GNU/Linux** - Segundo Cobbaut (2013) não há nenhuma empresa por traz de Debian, apenas centenas de programadores e elegem o chefe do projeto em 2 em 2 anos.

²⁰Disponível em [<http://www.ntchosting.com/fedora-linux-based-operating-system.html>], consultado em 16/07/2013

²¹ Disponível em [<http://www.redhat.com/products/enterprise-linux/>], consultado em 16/07/2013

- **Ubuntu** - Tudo começou em 2004 quando Canonical disponibilizou discos (CD) com Ubuntu grátis, rapidamente Ubuntu tornou se popular a muitos utilizadores trocaram Windows por Ubuntu (Cobbaut, 2013).
- **OpenSUSE** - É um sistema operativo baseado no Linux, desenvolvida pela comunidade openSUSE de forma gratuita, depois foi adquirida pela Novell em 2004 (Wikipédia, 2013d). SuSE começou a publicar a sua distribuição em Inglês em 1997, que abriu um escritório em Oakland, Califórnia naquela época (McCallister, 2008).
- **Gentoo** - ²² É um sistema operativo *Open Source* baseado em Linux ou FreeBSD que pode ser otimizado e personalizado para qualquer aplicação ou necessidade automática. Com o desenvolvimento da tecnologia chamada Portage (sistema de distribuição de software para Gentoo), o Gentoo pode também se tornar um servidor seguro, estação de trabalho de desenvolvimento, desktop profissional, solução integrada. (Negus, 2010) acrescenta ainda que a distribuição Gentoo é altamente personalizável.
- **Yellow Dog** – ²³ É um sistema *Open Source* baseado no kernel Linux, lançado na Primavera de 1999 e desenvolvido por Fixtars. Yellow Dog é um sistema para computação de alto desempenho em arquiteturas de computadores com processadores multi-core.
- **Slackware** –²⁴ Atualmente na versão 14.1 Slackware foi um dos primeiros sistemas operativos construídos acima do Kernel Linux e é o mais antigo sendo mantido, criado em 1993 por Patrick Volkerding. Sistema caracterizado pela estabilidade de *design* e simplicidade.

7 Considerações finais

Neste Capítulo vimos principais tecnologias *Open Source*, mas para falar dessas tecnologias elas foram divididas em categoria onde fez-se referência as mais utilizadas, onde as categorias

²² Disponível em [<http://www.gentoo.org/main/en/about.xml>], consultado em 16/07/2013

²³ Disponível em [http://en.wikipedia.org/wiki/Yellow_Dog_Linux], consultado em 16/07/2013

referidas foram Sistemas Operativo, Base de Dados, Editores, Servidores, IDEs, *Web browser*.

Dos software visto acima, vimos as principais características dos mesmo, onde pode-se reparar que segundo alguns autores esses softwares *Open Source* podem ter igual ou melhor qualidade que os softwares proprietário, como por exemplo entre a GIMP e o Photoshop.

Vimos também quadros estatísticos da utilização dessas tecnologias juntamente com softwares proprietários onde algumas dessas tecnologias representam superioridade em relação as suas alternativas proprietárias, como no caso do Servido Apache que domina o mercado de servidores com uma percentagem superior a 50%.

²⁴ Disponível em [<http://en.wikipedia.org/wiki/Slackware>], consultado em 16/07/2013

Capítulo 3: Estudo de Caso

1 Introdução

Para realizar este estudo de caso foi elaborado dois questionários, um direcionado para trabalhadores das empresas de desenvolvimento de software e o segundo para ser respondido por chefes de laboratórios das instituições de ensino superior na Cidade da Praia. A escolha desses dois tipos de instituição foi devido ao facto das universidades serem onde preparamos para integrar no mercado de trabalho, é onde aprendemos a trabalhar, nesse caso se nas universidades trabalhamos só com softwares proprietário e sairmos das universidades para trabalhar e encontra grande percentagem de *softwares Open Source* na empresa onde vamos trabalhar, pois constituiria um grande problema, isto leva-nos a crer que devia haver uma harmonia entre os softwares utilizados nas empresas e nas universidades.

Logo depois de aplicar os questionários, online, pois assim fica mais fácil aplicá-los online onde enviamos link dos questionário e os inquiridos respondem sem a necessidade de dirigir até a empresa fez-se o tratamento dos dados.

Ferramentas de apoio:

Durante a realização deste trabalho utilizou-se as seguintes ferramentas para que os resultados fossem mais claro possível:

- Google Drive;
- Excel;
- IBM SPSS Statistics;

O Google Drive foi utilizado para recolher dados, ou seja, criar os questionários a serem respondidos por parte das empresas inquiridos. Podia-se perfeitamente fazer os questionários em papel, mas foi preferida o Google Drive, pois trazia algumas vantagens como menor custo, dados mais fácil de ser tratado, tempo gasto na transferência dados de papéis para computadores.

O Excel foi utilizado para fazer correção dos dados, visto que no campo “outro” do questionário, uma pessoa pode responder “Sim” um outro pode responder “S” por exemplo ambos quer dizer a mesma coisa mas não está escrito igual.

Porém os dados do questionário num ficheiro Excel por si só não nos diz muita coisa, como é necessidade de tratar esses dados de modo a ficar mais legíveis e permitir tirar algumas conclusões, e nesse caso nada melhor que uma ferramenta dedicado a esta tarefa como o IBM SPSS Statistics, que permite inserir as respostas dadas pelo utilizadores e gerar gráfico à partir do mesmo, cruzar dados e outras detalhes.

1.1 Instituições do ensino superior que participaram

Como foi referido acima realizamos um questionário que foi aplicado nas universidades, visto que elas são muito importantes porque é nas instituições que preparam futuro profissionais de uma qualquer país e nesse caso realizamos o questionário com o objetivo de sabe qual é a percentagem de utilização de softwares *Open Source*, que problema é que enfrentam na utilização dos softwares *Open Sources*, se problemas relacionados com os softwares *Open Sources* são resolvidos por técnicos da instituições ou por fontes externos.

Em suma pode-se assim dizer que se algo tem de mudar as instituições de ensino superior são locais de muita importância para realizar esta mudança.

Instituições de ensino superior inqueridas:

- Universidade de Cabo Verde (Uni-CV);

- Universidade Jean Piaget de Cabo Verde;
- ISCCE;

As pessoas que responderam os questionários nas instituições de ensino superior a maioria são gestores dos laboratório e gestores de tecnologias de informação, porque é essas pessoas que fazem a gestão do laboratório dessas instituições (adquirir, instalar, fazer manutenção etc.) e acredita-se que estes nos dão uma resposta melhor.

1.2 Empresas base tecnológica que participaram

Realizou-se também um questionário direcionado para empresas, pois acima referimos que as instituições de ensino superior são de extrema importância, sabendo que prepara os alunos para mais tarde entrarem nas empresas, já as empresas não precisa-se dizer que são importante para o desenvolvimento de qualquer país. No entanto, estes dois questionários permite-nos saber se as empresas de base tecnológica de software e instituições de ensino superior da Cidade da Praia estão preparadas para uma política rigorosa de punição da violação dos direitos de autores como o que ocorre em certos países.

Empresas:

- Mgo Consulting;
- Garantia Seguros;
- NOSI;
- Compta;
- Prime;
- Mensagens e Soluções de Marketing;
- Zing Developers;
- CETIC Lda/ACONTECE;
- Sisp;
- ElseIt;
- Sismedia;

- Primacis;
- Bonako;
- Tei;
- TLC;
- Informática de Cabo Verde.

Dos empregados das empresas que participaram do questionário, 31,25% são analistas, 31,25% são programadores, 18,75% são Gerentes 6,25% são diretores do departamento técnico, 6,25% são administradores do sistema, no entanto maioria dos participantes são analistas e programadores, isto quer dizer que maioria dos inquiridos entende da matéria (*Open Source*).

2 Softwares Open Sources em Cabo Verde

No que se refere a sistemas operativos em Cabo Verde, destaca mais os sistemas operativos proprietário, o Linux fica numa posição baixíssima de uma forma geral, ou melhor dizer não só em empresas base tecnológica e instituições de ensino superior, mas sim em Cabo Verde inteiro. Os gastos nunca foram uma vantagem para softwares *Open Source* num país como Cabo Verde, que segundo (I. B. da Rosa & Mendes, 2010), onde maiorias das pessoas nunca pagaram licença de nenhum softwares que costumam utilizar. É evidente que as estruturas dos sistemas de informação cabo-verdiana estão fortemente dependente de softwares proprietários, olhando para o gráfico da figura abaixo que mostra a utilização de sistemas operativos em Cabo Verde.

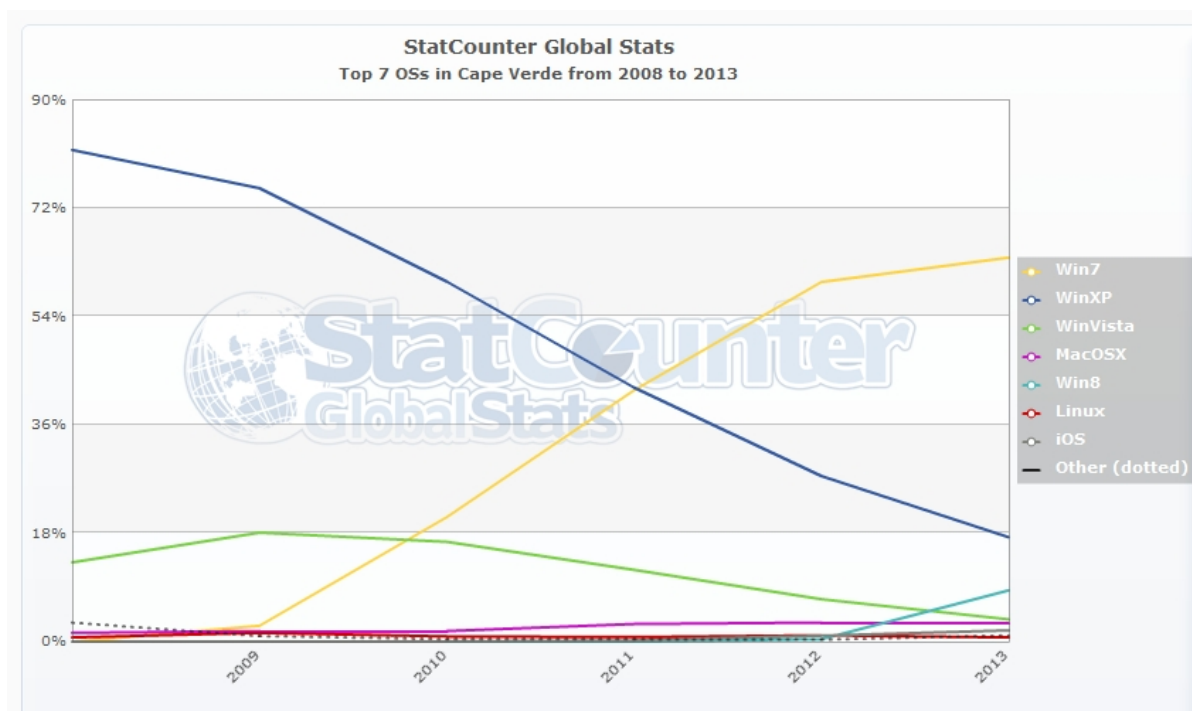


Figura 15: Índice de utilização de sistemas operativo em Cabo Verde (Stats, 2013b)

Ao verificar gráfico da figura 15, pode-se ver que o mercado de Cabo Verde é um mercado fortemente dominado pelos softwares proprietário na categoria sistemas operativos, onde o sistema operativo mais utilizado no momento é o Windows 7 com 60 e poucos por cento, em segundo lugar se encontra o Windows XP com aproximadamente 17% do mercado, ambos proprietários, o Linux único sistema operativo *Open Source* presente na lista se encontra em 6º lugar com aproximadamente 1% do mercado, ou seja muito baixo.

Pode-se reparar um facto interessantíssimo no gráfico, no ano 2011 houve uma troca entre os o sistema operativo a liderar a lista, mas entre softwares *Closed Source* e da mesma empresa (Windows), visto que de 2008 a início de 2011 o líder do mercado era Windows XP e no início de 2011 passou a ser Windows 7 ambos softwares da Microsoft e o estranho no meio de tudo isso é que o Linux em nenhum dos anos ultrapassou 2%.

No início de ambos os questionários, foram perguntado para os inquiridos em que circunstâncias utilizavam os softwares *Open Sources*, 75% responderam que “sempre que haja uma alternativa *Open Source*”, 25% responderam que “se a alternativa *Open Source* for melhor que o software proprietário”, mas no decorrer do questionário percebe-se que há uma certa contradição nas respostas.

2.1 Nas empresas de base tecnológica

2.1.1 Servidores

Analisando os dados pode-se constatar que a *Web* é muito utilizado pelas empresas de base tecnológica da Cidade da Praia, pois maioria dos inquiridos confessaram utilizarem servidores *Web* nas suas empresas, no entanto surgiu a questão “qual o servidor *Web* é que utilizam?” e podemos conferir nos gráficos das figuras abaixo (ver a figura 16 e 17):

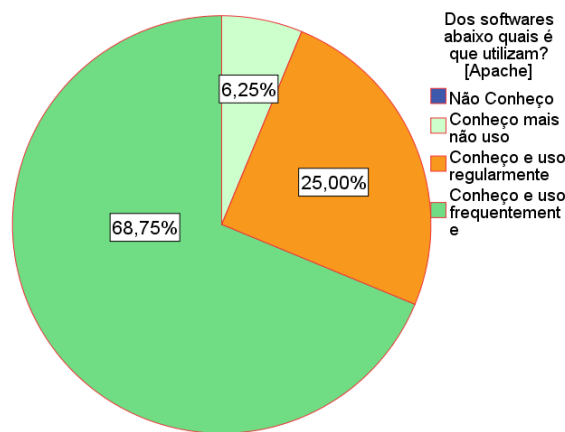


Figura 16: Percentagem da utilização do servidor apache nas empresas de base tecnológica na Cidade da Praia

O gráfico da figura 13 indica-nos a percentagem de utilização do Servidor Apache pelas empresas de base tecnológica na Cidade da Praia. Consoante o gráfico desta figura pode-se ver que 66,67% das empresas inquiridas utilizam frequentemente o servidor Apache, 26,67% utilizam regularmente e 6,67% alegam conhecerem, mas não o utilizam.

Ao comparar o Apache com o principal rival o IIS da Microsoft obtiveram o seguinte resultado:

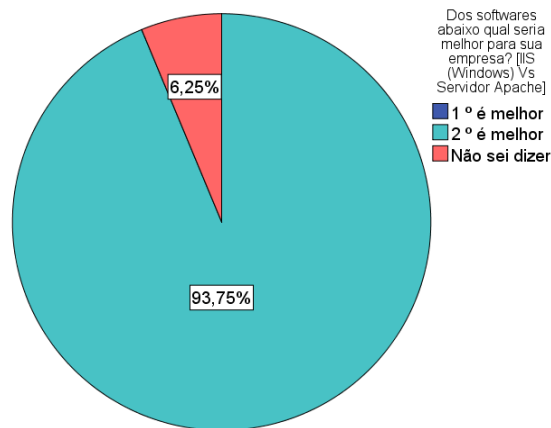


Figura 17: Servidor Apache vs. IIS Windows nas empresas de base tecnológica na Cidade da Praia

Na figura 17, apresenta o índice dos servidores que os inqueridos consideram ser melhor para as suas empresas, onde obtiveram o seguinte: 93,33% consideram o Servidor Apache como sendo o melhor para as suas empresas e 6,67% alegam não saberem qual é o melhor entre o Apache e IIS Windows.

Outro aspeto que se verificou foi os dados obtidos nesse questionário sobre a comparação do Servidor Apache e IIS Windows, e o índice da utilização do apache a nível mundial não difere muito, pois a nível mundial o Servidor Apache possui mais de 60% do mercado.

2.1.2 Browser

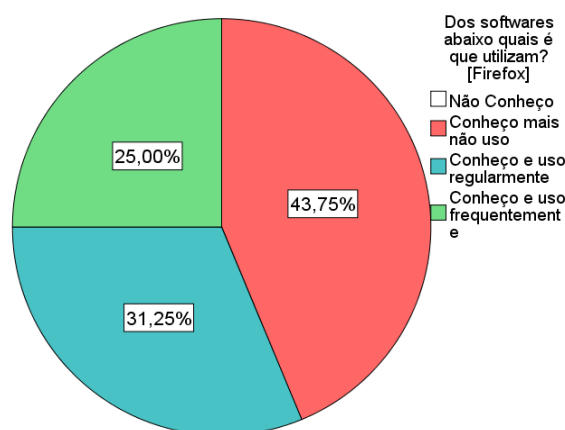


Figura 18: Índice de utilização do Mozilla Firefox nas empresas de base tecnológica na Cidade da Praia

Analisando o gráfico da figura 18, pode-se constatar que $\frac{1}{4}$ dos inquiridos é que são utilizadores fiéis do Mozilla Firefox, o outro $\frac{3}{4}$ encontra-se dividido entre o Mozilla e outros *browsers*. Dentro desse $\frac{3}{4}$, 31,25% utilizam o Mozilla Firefox regularmente e 43,75% conhecem o Mozilla Firefox, mas não utilizam.

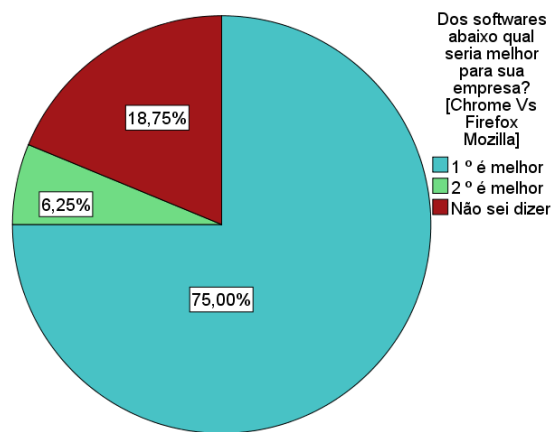


Figura 19: Google Chrome vs. Mozilla nas empresas de base tecnológica na Cidade da Praia

Ao analisar a figura 19, pode-se ver que praticamente $\frac{3}{4}$ dos inquiridos acham que o Google Chrome seria melhor ou é melhor para a sua empresa, 6,25% acham que o Mozilla Firefox seria melhor para a sua empresa e 18,75% não sabem dizer entre os dois navegadores qual seria melhor para suas empresas.

2.1.3 Sistemas Operativos

No que se refere aos sistemas operativos, segundo o questionário aplicado pode-se ver claramente que o Windows é o sistema operativo que domina o mercado sabendo que 100% dos inquiridos confessaram serem dependente do Windows para o cumprimento das tarefas do dia-a-dia.

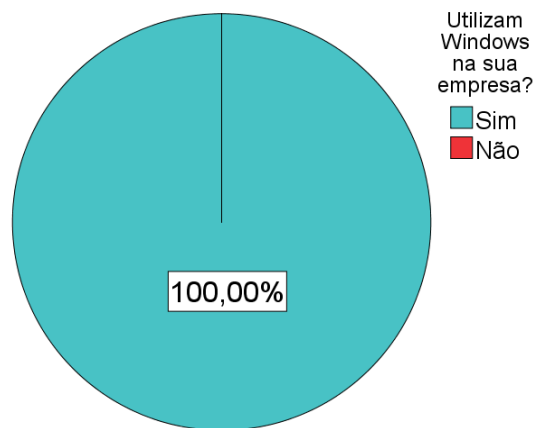


Figura 20: Utilização do SO Windows pelas empresas de base tecnológica na Cidade da Praia

Se comparamos a utilização do Windows com a utilização do Linux, pode-se ver que nem 50% dos inquiridos utilizam Linux, e isto ajuda-nos a não estranhar a figura 15 que é uma estatística global onde maiorias das pessoas não são profissionais da informática em comparação com as pessoas das empresas inqueridas, em que maiorias são analista, programadores, entre outros.

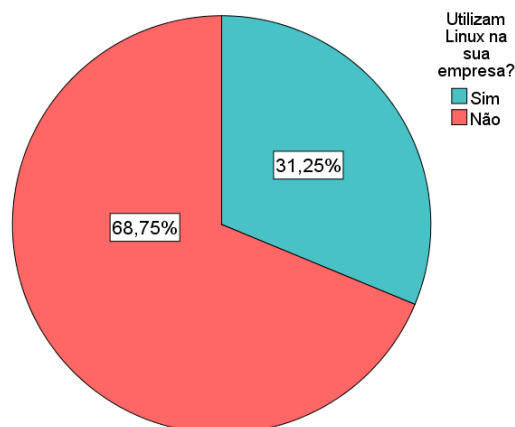


Figura 21: Utilização do Linux pelas empresas de base tecnológica na Cidade da Praia.

Segundo os critérios que foram apresentado aos utilizadores, o “preço” é o critério mais valorizado no momento da escolha do sistema operativo acumulando um total de aproximadamente 33%, em segundo lugar fica a “segurança” com um total de aproximadamente 20% e por fim vem os restantes critérios como a “facilidade” com 13,33%, “software veio pré-instalado” com 13,33% também, “disponibilidade” com 6,67%, “suporte” com 6,67% também e “depende” com 6,67% também. Contudo sabendo que o Linux é um

dos softwares de referência da *Open Source*, como vimos no primeiro Capítulo 1, mas mesmo assim é pouco utilizado em comparação com o Windows por parte de estas empresas.

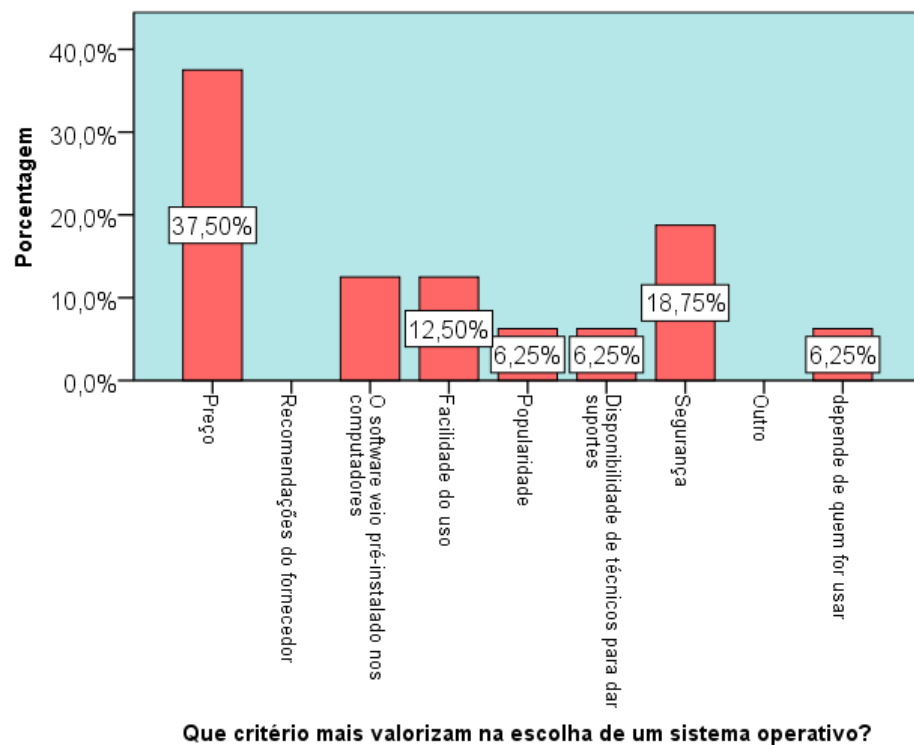


Figura 22: Critérios de avaliação de sistemas operativos nas empresas de base tecnológica na Cidade da Praia.

Mas agora surge uma pergunta muito interessantíssima que é, sendo o preço um dos critérios mais importante na escolha de um sistema operativo segundo os utilizadores que responderam o questionário, porque é que o Windows tem mais utilizadores que o Linux, sendo que o Windows é pago e o Linux é grátis? Segundo (I. B. da Rosa & Mendes, 2010) o preço do Windows não é muito acessível comparando com o salário de maioria dos utilizadores num país como Cabo Verde. Porém maioria das pessoas que utilizam Windows nunca pagaram pela licença, por isso em termos de preço o Linux e o Windows não têm diferença.

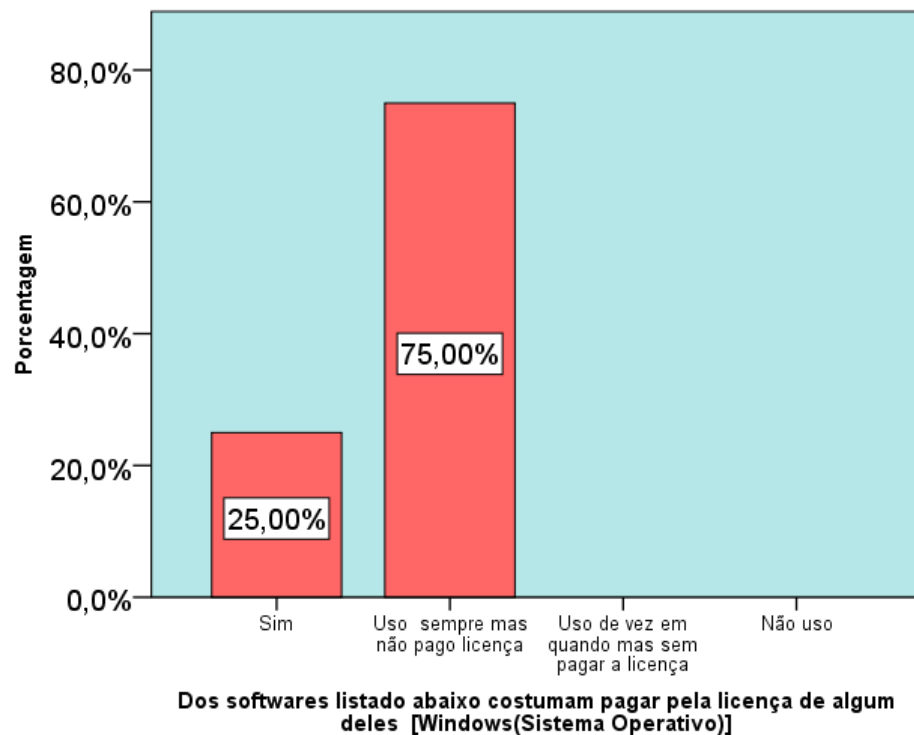


Figura 23:Índice de utilizadores que pagam pela licença do Windows nas empresas de base tecnológica na Cidade da Praia.

Analisando o gráfico da figura acima pode-se ver que, 25% dos inquiridos é que utilizam o Windows legalmente, os outros 75% não tem uma licença para utilização do Windows. Há autores como (Weerawarana et al., 2004) que afirmam que a diferença drásticas entre o índice da utilização dos softwares *Open Source* e *Closed Source* esta no facto das empresas proprietária têm uma campanha de marketing mais forte, também nem sempre são recursos da empresa, no caso de Windows logo ao surgir uma nova versão é publicitado e falado por muitos fóruns, *sites* relacionado com a informática e marketing, o que não acontece com o Ubuntu que é uma *distro* (distribuição Linux).

2.1.4 Editores

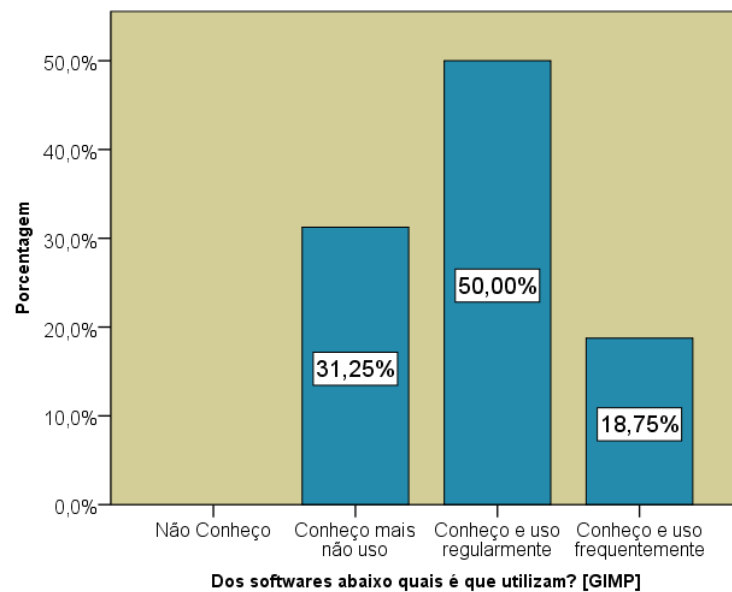


Figura 24:Índice de utilização do GIMP nas empresas de base tecnológica na Cidade da Praia.

Os dados mostram que todos os inquiridos conhecem o GIMP, 27% dos que conhecem, não o utilizam, 52% utilizam regularmente e só 20% dos inquiridos e que utilizam o editor frequentemente.

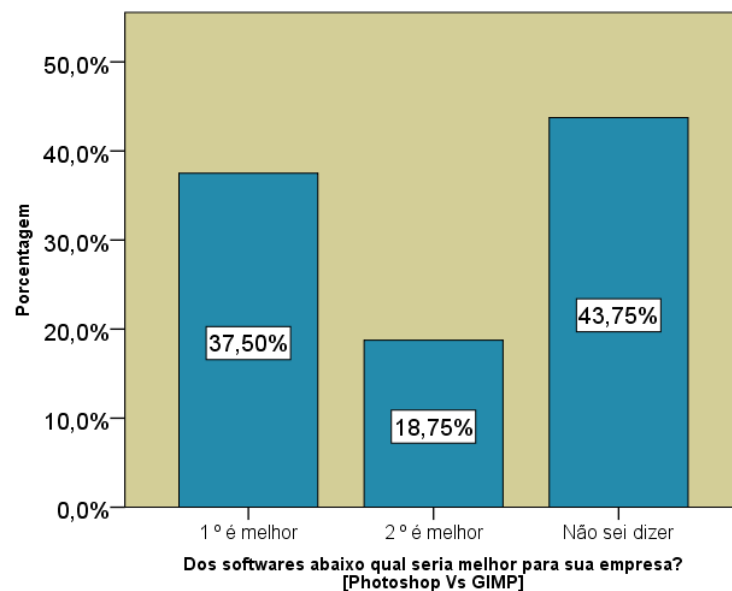


Figura 25: Photoshop vs. GIMP.

Ao serem questionados qual das plataformas é que seria melhor para as suas empresas, 33% responderam que o Photoshop seria melhor para as suas empresas, 20% disseram que o GIMP

seria melhor e 47% disseram que não sabiam avaliar qual seria melhor. Pode-se ver claramente que o Photoshop como sendo um software proprietário é muito mais utilizado que o GIMP que é *Open Source* e grátis.

No entanto é uma diferença bem grande entre a utilização dos dois softwares, porem segundo alguns autores possui características semelhantes ao Photoshop e com algumas vantagens como o preço sabendo que GIMP é grátis.

2.2 Nas instituições de ensino superior

As universidades são lugares onde os alunos são preparados para entrar no mercado de trabalho, onde os alunos ganham competências básicas exigidas pelo mercado, no entanto pode-se dizer que estes são pilares para a produtividade das empresas. Na realização dos questionários deparamos com um facto interessante, a dificuldade de manutenção referente a softwares *Open Source*, esse fenómeno podia melhorar se as instituições de ensino superior passassem a preparar os alunos para o tal e equipar os laboratórios das universidades com plataformas *Open Source*, como diz o ditado popular “É de pequeno é que se torce o pepino”.

Vimos acima que os softwares *Open Source* são pouco utilizado por parte das empresas de base tecnológica e isto não é preocupante, os administradores/chefes de cada empresa é que decidem o que utilizar nas suas empresas, mas o que é preocupante é que maioria dos softwares proprietário que essas empresas utilizam eles não pagam a referida licença, há empresas que não pagam pela licença de nenhum software que utilizam, e no caso de um dia houver mais rigor no que se refere ao pagamento das licenças dos softwares, essas empresas passarão por algum desconforto, provavelmente tem agora de pagar a licença dos principais softwares que utilizam ou mudar todos para *Open Source*, pois são mais barata e gastarem dinheiro para dar formações aos trabalhadores, ou melhor dizer esse facto causa algum desconforto as essas tais empresa.

2.2.1 Servidores

No entanto no que se refere a Servidores os resultados não se diferem tanto entre as instituições de ensino superior e empresas de base tecnológica.

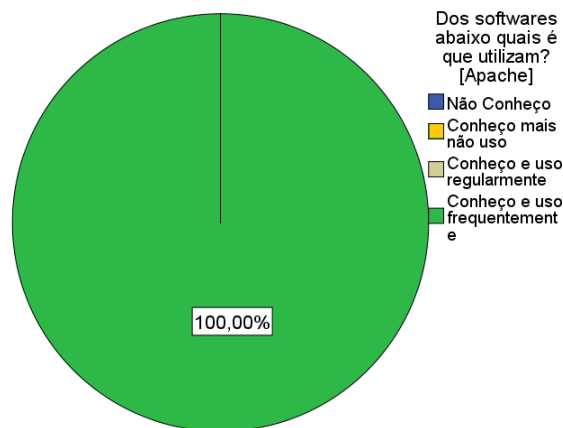


Figura 26: Utilização de servidor apache por parte das universidades na Cidade da Praia.

Fica comprovado que o Apache é o servidor Web mais utilizado por parte das instituições de ensino superior na Cidade da Praia e em empresas de base tecnológica, pode-se reparar na figura 26 que 100% dos inquiridos confessaram que utilizam o servidor apache frequentemente.

2.2.2 Browser

No que diz respeito a navegadores/*browsers* os dados obtidos nas universidades difere com os dados obtidos pelas empresas de base tecnológica, visto que nas empresas maioria consideram que o Google Chrome seria melhor para suas empresas enquanto que nas universidades mais de 65% dos inquiridos consideram que o Mozilla Firefox seria melhor para eles. O que também contradiz com os dados da utilização a nível mundial, em que a maioria dos utilizadores utilizam o Google Chrome como seu navegador principal, pode-se conferir na página 56 figura 8, no entanto não só contradiz com os dados da utilização mundial e os dados das empresas inquiridas mas também com os dados de Cabo Verde que pode-se conferir na figura a baixo.

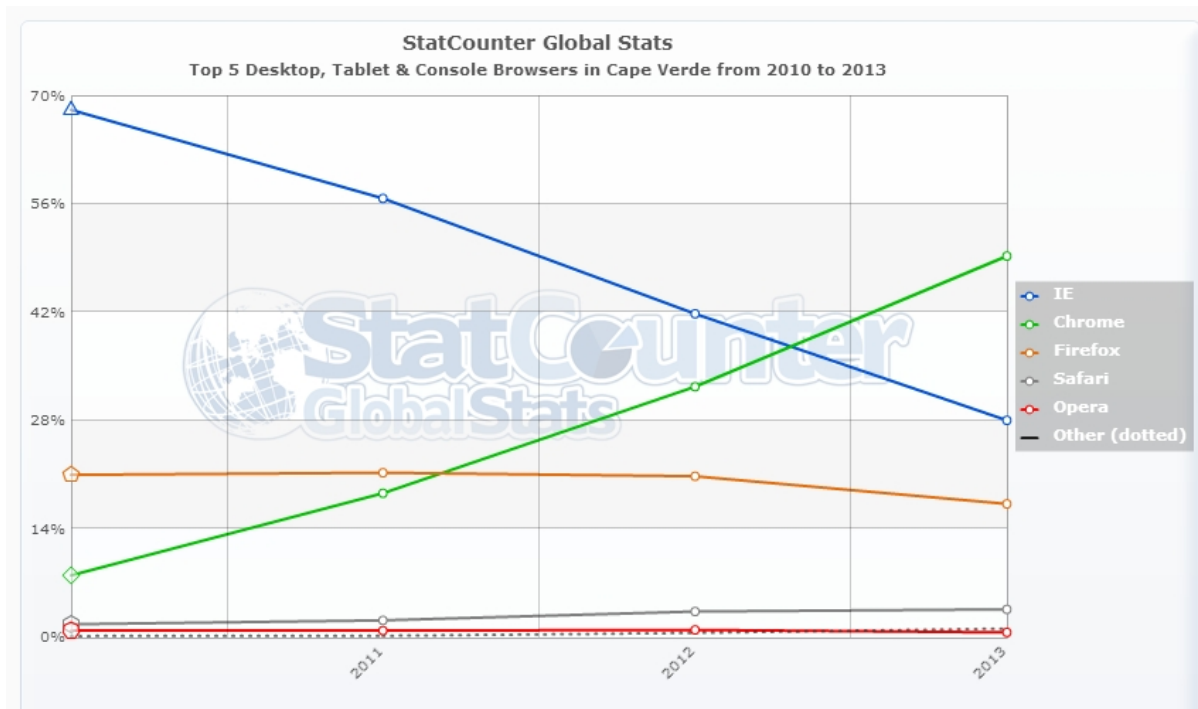


Figura 27: Top 5 browser mais utilizados em Cabo Verde de 2010 à 2013.

Se repararmos na figura 27, pode-se constatar claramente o progresso fantástico do Google Chrome nestes últimos 3 anos, que no entanto não é um *browser Open Source*, mas é gratuito, no entanto o Internet Explorer nesses últimos 3 anos esteve a perder o número de utilizadores que também não é *Open Source*. Agora vamos ao que é o foco deste trabalho, os *browsers Open Source* e o único da lista é o Mozilla Firefox que nos últimos 3 anos quase não teve alterações, com aproximadamente 20%, tendo uma pequena queda não muito brusca em comparação com o Internet Explorer que já vem incluído nos sistemas operativo da Microsoft (Windows).

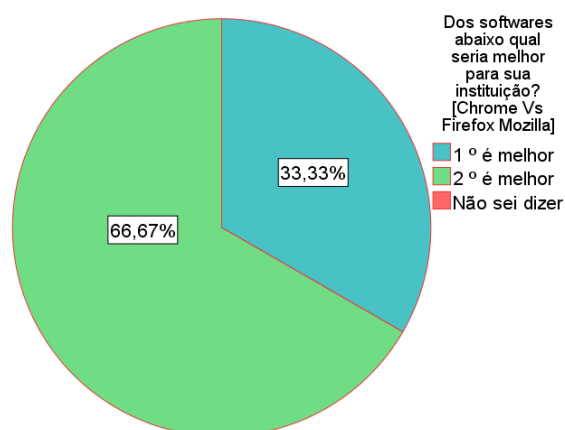


Figura 28: Google Chrome vs. Mozilla Firefox nas universidades da Cidade da Praia.

No entanto pode-se ver na figura acima, que nas universidades da Cidade da Praia a realidade é diferente das empresas de base tecnológica e Cabo Verde em geral, visto que 66,67% dos inquiridos afirmam que o Mozilla Firefox é melhor para as suas instituições, enquanto que só 33,33% é que afirmam que o Google Chrome seria melhor para as suas instituições.

2.2.3 Sistemas Operativos

Durante o decorrer deste trabalho, vimos que os computadores não fazem quase nada sem um sistema operativo, no entanto por tudo quanto é lado que há computadores há sistemas operativo quer proprietário quer *Open Source*, no entanto ao serem questionados os responsáveis dos laboratórios de informática “que sistemas operativos é que utilizam nos laboratórios”, vê-se que os sistemas operativos proprietário é que dominam no entanto Windows, visto que todos os inquiridos confessaram utiliza-lo e pode-se ver os resultados na Figura abaixo:

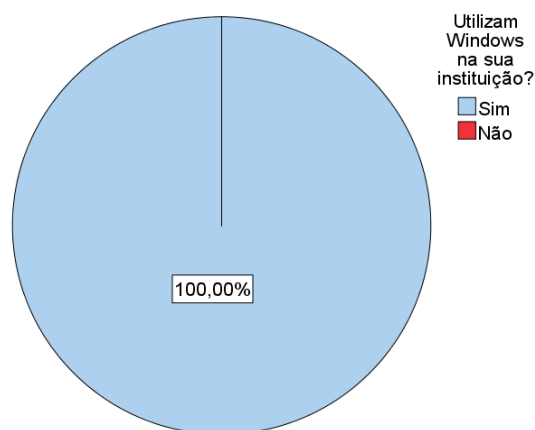


Figura 29:Índice de utilização do Windows por parte das instituições na Cidade da Praia.

Ao ver a figura 29, pode-se ver que todas as instituições de ensino superior na Cidade da Praia que participaram do inquérito utilizam Windows. No entanto ao serem questionados se “utilizam Linux nos seus laboratórios de informática”, só 66,67% é utilizam, 33,33% não utilizam. No entanto o Mac OS nenhum dos inquiridos utilizam e Solaris também nenhum dos inquiridos utilizam, em suma pode-se ver que o Windows e o Linux são os principais sistemas operativos que são utilizado nos laboratórios de informática dessas instituições.

2.2.4 Editores

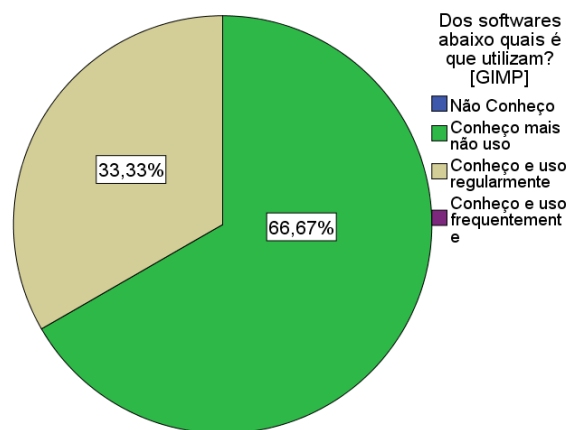


Figura 30: Índice de utilização do GIMP pelas instituições na Cidade da Praia.

Ao analisarmos a figura acima, pode-se ver claramente que nenhuma das instituições de ensino utilizam o GIMP com frequência, ou seja essas instituições utilizam o GIMP ocasionalmente. No entanto essa fraca utilização do GIMP nessas instituições pode estar relacionado ao seguinte facto, ao serem questionado os responsáveis dos laboratórios se pagam pela licença do Adobe Photoshop, 100% dos inquiridos afirmaram utilizarem sempre mas sem pagar a devida licença do mesmo, se não pagam pela licença do principal alternativa do GIMP que é proprietário isto poderá estar ligado a fraca utilização do GIMP.

3 Outras informações obtidas

Na realização dos questionários verificaram também as seguintes informações que não foram mencionadas acima, mas que acreditam ser importante também:

softwares *Open Sources* mais populares na lista dos softwares presente no questionário aplicado às empresas de base tecnológica e instituições de ensino superior na Cidade da Praia:

- Linux (qualquer sistemas operativo que utiliza o Kernel Linux);
- Servidor Apache;
- Mozilla Firefox;
- Eclipse;
- Netbeans.

Softwares *Open Sources* menos populares na lista dos softwares presente no questionário aplicado às empresas de base tecnológica e instituições de ensino superior na Cidade da Praia:

- Tellico;
- Koffice;
- Grant Project;
- Inkscape;
- Scribus;
- Kompozer;
- Audacity;
- Image Magick;
- AbiWord;
- Neo Office.

No entanto pode-se ver claramente que a maioria dos softwares *Open Sources* que foram apresentados nos questionários são pouco utilizado pelas empresas de base tecnológica e instituições de ensino superior, de entre os poucos utilizados há uma percentagem bem grande que os inquiridos alegam não conhecerem como no caso dos citados acima.

Pode-se ver também que segundo as informações extraída nos dados dos questionários, que nestas instituições inquirida, que há uma fraca participação dos mesmos nas comunidades *Open Source*. Sendo que das pessoas pertencente a empresas inquiridas, 68,75% delas não fazem parte de nenhuma comunidade do software *Open Source*, só 31,25% é que pertencem a alguma comunidade. A fraca participação dos utilizadores inquiridos nas comunidades *Open Source* pode estar relacionado com a dominação dos softwares proprietários nessas instituições o que diminui o interesse dos mesmos em participar nas comunidades *Open Source*.

No que se refere aos critérios de escolha de um software para utilização nessas duas instituições, os critérios mais importantes que os inquiridos indicaram são:

- Preço;

- O software veio pré instalado nos computadores;
- Segurança.

Os critérios menos importantes para os inquiridos são:

- Facilidade da utilização;
- Popularidade;
- Disponibilidade de técnicos para dar suporte.

No que diz respeito a resolução de problema relacionado com os softwares *Open Source* das empresas de base tecnológica inquiridas, os dados indicam que 12,50% dos inquiridos resolvem o problema sempre, 56,24% confessaram resolverem o problema geralmente e 31,25% confessaram resolverem o problema as vezes, no entanto nenhuma das empresas responderam nunca. No entanto as instituições de ensino superior mostraram serem um pouco menos confiante na resolução dos problemas, visto que nenhuma delas responderam que sempre que haver um problema com um software *Open Source* este é resolvido, 66,67% responderam geralmente e 33,3% responderam as vezes.

As questões de vantagens e desvantagens também fazem parte do questionário, onde é apresentado um conjunto de vantagens para o inquirido marcar se este é realmente uma vantagem que o software dispõe para sua empresa e o mesmo acontece com a desvantagem e obtiveram os seguintes resultados:

- Vantagem competitiva- 56,25% dos inquiridos acreditam que utilizando softwares *Open Source* tornam as suas empresas mais competitiva e 43,75% discordam;
- Baixo custo – 93,75% dos inquiridos acreditam que utilizando softwares *Open Source* poupam consideravelmente o dinheiro da empresa e 6,25% discordam;
- Mais acessível – 68,75% dos inquiridos acreditam que os softwares *Open Source* são mais acessíveis para as suas empresas e 31,25% nem tanto;
- Melhor adaptação as necessidades – 62,50% dos inquiridos acreditam que os softwares seriam mais fácil de adaptar as necessidades das suas empresas e 37,50% discordam;
- Melhor disponibilidade – 25% dos inquiridos acham que o software *Open Source* tem vantagem no que diz respeito a disponibilidade e 75% discordam;

- Mais rapidez na realização das tarefas – 31,25% acham que a rapidez é uma vantagem que o software *Open Source* trás para suas empresa e 68,75 discordam.

	Empresas de base tecnológica		Instituições de ensino superior	
	Sim	Não	Sim	Não
Preço	93,75 %	6,25 %	100,0 %	0,00 %
Mais bem testado	18,75 %	81,25 %	0,00 %	100,00 %
Mais personalizável	81,25 %	18,75 %	33,33 %	66,67 %
Relatório de erro mais completo	18,75 %	81,25 %	33,33 %	66,67 %
Mais estável	31,25 %	68,75 %	0,00 %	100,0 %
Mais Leve (Combina com hardwares de baixo)	56,25 %	43,75 %	0,00 %	100,0 %
Melhor qualidade	25,00 %	75,00 %	33,33 %	66,67 %
Outro				

Tabela 6: Vantagens do Open Source sobre o Closed Source segundo os inquiridos.

A tabela acima indica-nos um conjunto de vantagens ou seja propostas e a percentagem de pessoas que acreditam que as tais propostas são de facto uma vantagens e a percentagens das pessoas que não acreditam nas empresas de base tecnológica e nas instituições de ensino superior.

Das principais vantagens que os inquiridos apontaram sobre os softwares *Open Source* são preço mais acessível, mais personalizável e mais leve.

	Empresas de base tecnológica		Instituições de ensino superior	
	Sim	Não	Sim	Não
Pouca documentação	75,00 %	25,00 %	33,33 %	66,67 %
Difícil de dar manutenção	31,25 %	68,75 %	0,00 %	100,00 %
Difícil de utilizar	31,25 %	68,75 %	33,33 %	66,67 %
Interface não <i>user-friendly</i>	37,50 %	62,50 %	33,33 %	66,67 %
Pouca usabilidade;	18,75 %	81,25 %	0,00 %	100,0 %
Incompatibilidade dos programas	37,50 %	62,50 %	66,67 %	33,33 %
Pouca garantia	25,00 %	75,00 %	33,33 %	66,67 %
Mudança de versão rápida	62,50 %	37,50 %	0,00 %	100,00 %
Baixa qualidade	0,00 %	100,00 %	0,00 %	100,00 %
Outra:				

Tabela 7: Desvantagens do Open Source em relação ao Closed Source segundo os inquiridos.

A tabela acima indica-nos um conjunto de desvantagens ou seja propostas e a percentagem de pessoas que acreditam que as tais propostas são de facto uma vantagem e a percentagens das pessoas que não acreditam nas empresas de base tecnológica e nas instituições de ensino superior.

Dos principais problemas que pessoas mais se queixam sobre os softwares *Open Sources* são a pouca documentação e mudança de versão rápida.

Conclusão

No decorrer deste trabalho fica claro que o desenvolvimento de uma país esta fortemente ligado as tecnologias de informação, e essas tecnologias são movido a base de softwares. Weerawarana et al. (2004:p.30) afirma ainda que (...) “a definição e adoção de normas, o investimento em tecnologia, treinamento e desenvolvimento de habilidades pode ter graves consequências para o futuro bem-estar de seus povos”. Pode se perfeitamente dizer que os softwares são fundamentais para o desenvolvimento de um país, mas estes nem sempre é de graça como vê-se no início do trabalho.

O objetivo geral deste trabalho era saber quais vantagens e oportunidades que os softwares *Open Sources* oferecem para países subdesenvolvidos como Cabo Verde, e saber o índice da utilização desta tecnologia por parte das empresas de base tecnológica e instituições de ensino superior na Cidade da Praia.

De acordo com os dados obtidos das pesquisas e do questionário pode-se ver que os softwares *Open Sources* oferecem bastantes vantagem e oportunidades para países em via desenvolvimento, como por exemplo ele ajuda a diminuir a enorme distância existente entre os pises desenvolvidos e países em desenvolvimento como no caso de Cabo Verde e Estados Unidos de América, o que softwares *Closed Sources* não oferecem, sabendo que estes não disponibilizam código fonte e torna-se difícil de estuda-los o que de uma certa forma gera

dependência segundo Richar Stallman no caso de uma empresa possuir um IDE *Closed Source* por exemplo e querer fazer alguma alteração este tem de pedir ao fornecedor. Os dados mostram que a utilização de softwares *Open Source* por parte das empresas que analisamos é muito baixo em comparação com a *Closed Source*.

Descobrimos também que os softwares *Open Source* possuem vantagens que não se aplicam como no caso do preço sabendo que maioria das pessoas/utilizadores inquiridos não pagam pelas licenças dos softwares proprietário que de uma forma geral são mais caros que os softwares *Open Source* desta forma o preço em Cabo Verde nunca foi uma vantagem para softwares *Open Source*.

No que se refere ao caso prático fica claro que muitas vezes os inquiridos respondem certas pergunta e contrariaram com outras respostas, por exemplo comparação entre softwares *Open Source* e *Closed Source*, maioria dos inquiridos respondem que os softwares *Open Sources* são melhores que os softwares proprietário, mas não utilizam os softwares *Open Source* que consideram melhor que o software proprietário. Um fenómeno também que verificamos foi o seguinte grande partes doas softwares *Open Sources* são pouco utilizado e outros até pouco conhecido por parte dos inquiridos.

Perante os dados analisados dos resultados da aplicação dos questionários às empresas de base tecnológica e das instituições de ensino superior deixamos abaixo algumas recomendações:

- Preparar os alunos já nas universidades a acostumarem com tecnologias *Open Source* de modo a estarem mais preparado para resolver problemas com softwares *Open Source*, isto ajuda as empresas onde futuramente irão trabalhar;
- Preparar as empresas para abraçaram o movimento *Open Source* sabendo que esses oferecem muitas vantagens para países como Cabo Verde, ou seja países em via desenvolvimento, visto que estes em comparação com países desenvolvidos não possuem recursos suficiente para gastarem com licença de softwares proprietário, como vê-se os softwares proprietário geralmente são mais caras que os softwares *Open Sources*;

- Empresas precisam dar pequenas formações aos técnicos e pessoas de manutenção de modo a permitir que esses ganham maior capacidade de dar manutenção nos softwares *Open Source*;
- Começar a utilizar softwares *Open Source* com mais frequência que por exemplo um dia houver maior rigorosidade no que se refere ao pagamento das licenças dos softwares proprietários sabendo que quase não pagam licença;
- As universidades precisam publicitar os softwares *Open Sources* para que futuramente evitar problemas com pagamento de licença se no caso o pagamento das licenças começaram a ser aplicadas com mais rigor, já ai encontram se boa parte dos alunos habituado com os softwares *Open Sources*;
- No caso das empresas que não pagam a licença pelos motivos financeiros, e por outro lado utilizam Windows e outros softwares proprietário no dia-a-dia, seria melhor reestruturar a empresa utilizando maioritariamente softwares *Open Source*, visto que softwares *Open Sources* maioria das vezes são grátis;
- E para as empresas de base tecnológica e as instituições de ensino superior que queiram alguns dia lançar algum tipo de softwares *Open Source* tem que certificar que a documentação seja da melhor qualidade e que esteja ao alcance de todos, sabendo que a falta de documentação é um das coisas que os utilizadores mais se queixam;
- No entanto, para empresas ou comunidade que querem desenvolver softwares *Open Source*, antes de mais estes tem de pensar em como fica a interface a usabilidade, visto que este também é um fator que muitos inquiridos se queixam que muitos softwares *Open Sources* são défices de utilizar;
- Para pessoas/empresais que queiram algum dia desenvolver um software *Open Source* que engloba outras partes de softwares de terceiros antes de tudo este deve fazer um estudo para ver a compatibilidade das licenças destes softwares do terceiro com o que queira utilizar no seu projeto se são compatíveis, como vimos na página 38 nem todas as licenças são compatíveis.'

Há um ditado que diz que *projetos não são concluídos mas sim abandonado* e essa monografia não foge essa regra, no entanto este está sujeito a melhoramento futuramente.

Bibliografia

- AlternativeTo. (2014). Search Result for “Photoshop” - AlternativeTo.net. Retrieved May 29, 2014, from <http://alternativeto.net/SearchResult.aspx?search=Photoshop>
- ApacheFoundation. (2013a). Descrição das novas funcionalidades do Apache 2.0 - Servidor HTTP Apache. Retrieved September 05, 2013, from http://httpd.apache.org/docs/2.2/new_features_2_0.html
- ApacheFoundation. (2013b). Sobre o Projeto Servidor Apache HTTP - O Projeto Apache HTTP Server. Retrieved September 04, 2013, from https://httpd.apache.org/ABOUT_APACHE.html
- ApacheORG. (2013). QuickFix-WinXP.png (1024×768). Retrieved from <http://www.eclipse.org/screenshots/images/QuickFix-WinXP.png>
- Arjen Lentz, DuBois, P., & Hinz, S. (2006). *Manual de Referência do MySQL 4.1* (p. 919).
- Bacic, N. M. (2003). *O software livre como alternativa ao aprisionamento tecnologico imposto pelo software proprietario*. Universidade Estadual de Campinas . Instituto de Economia.
- Böck, H. (2009). *The Definitive Guide to NetBeans Platform* (p. 370).
- Bressan, M. M., & Filho, G. G. (2007). Software Livre Versus Software Proprietário:, 13.
- Campos, A. (2006). O que é software livre. Retrieved September 01, 2013, from <http://br-linux.org/linux/faq-softwarelivre>
- Ceruzzi, P. E. (2003). *A History of Modern Computing* (2nd ed., p. 460). London, Eng.; Cambridge, Mass.: Press, MIT.
- Cobbaut, P. (2013). *Linux Fundamentals* (p. 267).
- Colares, F. M. (2007). *ANÁLISE COMPARATIVA DE BANCO DE DADOS GRATUITOS*. FACULDADE LOURENÇO FILHO.
- Costa, J. A., & Melo, A. de S. (1999). *Dicionário da língua portuguesa* (p. 1794). Porto Editora.
- Engelfriet, A. (2003). Open innovation and Open source, 1–5.
- Faria, F. B., Lima, P. S. N., Dias, L. G., Silva, A. A., Costa, M. P., & Bittar, T. J. (2010). *Evolução e Principais Características do IDE Eclipse*. Universidade Federal de Goiás.
- Farias, C. G. S. de. (2011). *Estudo comparativo entre software livre e proprietário*. Faculdade de Tecnologia Da Zona Leste.
- Fernandes, A. S. (2008). *Tecnologias Livres de Desenvolvimento Web*.
- Ferraz, N. C. de T. (2002). *Vantagens Estratégicas do Software Livre para o Ambiente Corporativo*.

- Foundation, A. S. (2013). The Apache Software Foundation. Retrieved from <http://www.apache.org/licenses/>
- Fuggetta, A. (2003). Open source software—an evaluation. *Journal of Systems and Software*, 66(1), 77–90. doi:10.1016/S0164-1212(02)00065-1
- Garrels, M. (2008). *Introduction to Linux* (p. 223).
- Gentil, B. (1993). Usabilidade e Interação Homem-Computador, (1994), 49–73.
- Gimp. (2013). GIMP - About GIMP. Retrieved September 01, 2013, from <http://www.gimp.org/about/>
- GNU. (2013). GNU - Emacs. Retrieved from <http://www.gnu.org/software/emacs/>
- Group, T. P. G. D. (2013). *PostgreSQL 9.1.9 Documentation* (p. 2585).
- Gugik, G. (2009). Código Aberto e Software Livre não significam a mesma coisa! Retrieved from <http://www.tecmundo.com.br/linux/1739-codigo-aberto-e-software-livre-nao-significam-a-mesma-coisa-.htm>
- Inventta. (2013). Retrieved August 19, 2013, from <http://inventta.net/radar-inovacao/a-inovacao/>
- Iwata, R. R. (2009). *Software Livre x Software Proprietário e suas implicações econômicas e políticas . Software Livre x Software Proprietário e suas implicações econômicas e políticas .*
- Jandl, P. (2004). Notas Sobre Sistemas Operacionais.
- KernelOrg. (2013). About Linux Kernel. Retrieved August 28, 2013, from <https://www.kernel.org/linux.html>
- Kon, F. (2001). *O Software Aberto e a Questão Social*. Universidade de São Paulo. Retrieved from <http://gsd.ime.usp.br>
- Kylander, K., & Kylander, O. S. (1999). *GIMP User's Manual* (p. 924).
- Landim, W. (2011). A Evolução dos Navegadores : Mozilla Firefox, 3.
- Laurent, A. M. St. (2008). *Understanding Open Source and Free Software Licensing* (p. 201). O'Reilly.
- Lindberg, V. (2008). *Intellectual Property and Open Source* (p. 392). Beijing; Farnham: O'Reilly.
- LINFO. (2005). Proprietary Software Definition. Retrieved August 10, 2013, from <http://www.linfo.org/proprietary.html>
- Mandelbaum, A. (2012). Open Source Cost - The Google Trend That Has Not Wavered. Retrieved September 07, 2013, from <http://www.openlogic.com/blog/bid/230649/Open-Source-Cost-The-Google-Trend-That-Has-Not-Wavered>
- McCallister, M. (2008). *openSUSE Linux UNLEASHED* (p. 733). 800 East 96th Street, Indianapolis, Indiana 46240 USA: Sams Publishing.

- Meeker, H. J. (2008). *The Open Source Alternative : Understanding Risks and Leveraging Opportunities* (p. 301). New Jersey: John Wiley & Sons.
- Meireles, B. M. da C. (2009). *Open Source Software vs. Proprietary Software: An agent based modeling approach*. Universidade Tecnica de Lisboa.
- Michler, C. (2005). THE PROCUREMENT DECISION - “OPEN” OR “CLOSED” SOURCE SOFTWARE?
- Moglen, E. (2013). Free software and the death of copyright. Retrieved July 08, 2013, from <http://www.firstmonday.org/ojs/index.php/fm/article/view/684/594>
- Negus, C. (2010). *Linux Bible 2010 Edition* (p. 915). Wiley Publishing.
- NetBeans. (2013). A Brief History of NetBeans. Retrieved September 01, 2013, from <https://netbeans.org/about/history.html>
- NetBeansORG. (2011). web-projectlg.png (1400×842). Retrieved from https://netbeans.org/images_www/screenshots/6.0/web-projectlg.png
- Netcraft. (2013). 2013 | Netcraft. Retrieved September 02, 2013, from <http://news.netcraft.com/archives/2013/>
- Niederauer, J. (2006). *MySQL 5*. (R. PRATES, Ed.) (p. 15). Novatec Editora.
- PostgreSQL: About. (2014). Retrieved March 13, 2014, from <http://www.postgresql.org/about/>
- Potdar, V., & Chang, E. (2004). *Open Source and Closed Source Software Development Methodologies*. Curtin University of Technology.
- Ranking-DB-Engines. (2013). DB-Engines Ranking - popularity ranking of database management systems. Retrieved August 28, 2013, from <http://db-engines.com/en/ranking>
- Raymond, E. S. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3), 23–49. doi:10.1007/s12130-999-1026-0
- Reijswoud, V. Van, & Jager, A. De. (2008). *Free and open source software for development(Exploring expectations, achievements and the future)* (p. 117).
- Rob. (2013). What Is Linux | Linux.org. Retrieved September 09, 2013, from <http://www.linux.org/threads/what-is-linux.4076/>
- Rosa, I. B. da, & Mendes, J. M. (2010). FOSS and Software Expenditure in West Africa Draft Report.
- Rosa, I. B. O. da. (2010). *Costrução e utilização de bibliotecas digitais*. Universidade de Santiago de Compostela.
- Rosão, C., Figueiredo, R., & Mestre, T. (2010). Open Source, 11.
- Sabino, V., & Kon, F. (2009). *Licenças de Softwares Livre Historia e caracteristica* (pp. 1–36). São Paulo.

- Saltis, S. (2009). Open Source vs . Closed Source (Proprietary) Software.
- Santos, A. P. O. dos. (2012). *Aplicação de práticas de usabilidade ágil em software livre Ana*. Universidade de São Paulo.
- Schryen, G. (2009). Security of Open Source and Closed Source Software: An Empirical Comparison of Published Vulnerabilities, 13.
- Silva, F. J. da S. e. (2009). Software Livre : Conceitos , História e Impactos, 1–54.
- Silva, R. L. (2005). *MODELO DE AUTOMAÇÃO EM BIBLIOTECAS BASEADO NA FILOSOFIA OPEN SOURCE*. UNIVERSIDADE FEDERAL DO MARANHÃO.
- Soares, A. L. (2003). o caso do software de código aberto, 17.
- Stallman, R. (1981). EMACS The Extensible, Customizable Self-Documenting Display Editor.
- Stallman, R. (2007). Por que o Código Aberto não compartilha dos objetivos do Software Livre - Projeto GNU - Free Software Foundation (FSF). Retrieved July 07, 2013, from <http://www.gnu.org/philosophy/open-source-misses-the-point.html>
- Stats, S. G. (2013a). Top 5 Browsers from 2008 to 2013 | StatCounter Global Stats. Retrieved August 26, 2013, from <http://gs.statcounter.com/#browser-ww-yearly-2008-2013>
- Stats, S. G. (2013b). Top 7 Operating Systems from 2008 to 2013 | StatCounter Global Stats. Retrieved August 26, 2013, from <http://gs.statcounter.com/#os-ww-yearly-2008-2013>
- Steward, A. (2011). A Guide to Open Source Software for Australian Government Agencies, (June), 67.
- Taurion, C. (2004). História do Software - Mundo Digital. Retrieved August 25, 2013, from <http://noticias.uol.com.br/mundodigital/softwarelivre/2004/09/15/ult2449u3.jhtm>
- Webdevelopersnotes. (2013). What is web server? Retrieved August 30, 2013, from http://www.webdevelopersnotes.com/basics/what_is_web_server.php
- Weber, S. (2004). *The success of open source [...] [...]*. Cambridge, Mass. [u.a.: Harvard Univ. Press.
- Weerawarana, S., Weeratunga, J., & Sida. (2004). *Open Source in Developing Countries*. (Sida, Ed.) (p. 114). Stockholm.
- Wikimedia. (2013a). Copyleft symbol. Retrieved August 20, 2013, from <http://commons.wikimedia.org/wiki/File:Copyleft.svg>
- Wikimedia. (2013b). Copyright symbol. Retrieved August 20, 2013, from http://commons.wikimedia.org/wiki/File:Copyright_symbol_9.gif
- Wikimedia. (2013c). File:Tux.svg - Wikimedia Commons. Retrieved from <http://commons.wikimedia.org/wiki/File:Tux.svg>
- Wikipédia. (2013a). Banco de dados – Wikipédia, a enciclopédia livre. Retrieved August 29, 2013, from http://pt.wikipedia.org/wiki/Banco_de_dados

Wikipédia. (2013b). Firmware – Wikipédia, a enciclopédia livre. Retrieved August 27, 2013, from <http://pt.wikipedia.org/wiki/Firmware>

Wikipédia. (2013c). Linux – Wikipédia, a enciclopédia livre. Retrieved August 27, 2013, from <http://pt.wikipedia.org/wiki/Linux>

Wikipédia. (2013d). openSUSE – Wikipédia, a enciclopédia livre. Retrieved from <http://pt.wikipedia.org/wiki/OpenSUSE>

Wikipédia. (2013e).

Yeow, C. C. (2005). *Firefox Secrets: A need-to-know guide* (p. 297). SitePoint Pty.

A Anexo

A.1 Questionário às Empresas de base tecnológica

Qual é a empresa que representa? *

R:.....

Qual é a sua função? *

- ☐ - Programador ;
- ☐ - Analista;
- ☐ - *Designer*;
- ☐ - *Tester*;
- ☐ - Outro:.....;

Quando utilizam softwares *Open Source*? *

- ☐ - Sempre que haja uma alternativa *Open Source*;
- ☐ - Se a alternativa *Open Source* for melhor que softwares proprietário;
- ☐ - Se não tem softwares proprietário alternativo;

Indique os motivos da sua escolha: *

.....

Selecione algumas vantagens que *Open Source* traz para sua empresa: *

- ☐ - Vantagem competitiva;
- ☐ - Baixo custo;
- ☐ - Mais acessível;
- ☐ - Melhor adaptação às necessidades;
- ☐ - Melhor disponibilidade;
- ☐ - Mais rapidez na realização das tarefas;
- ☐ - Outro:.....;

Que sistema operativo vocês utilizam? *

- ☐ - Windows;
- ☐ - Linux;
- ☐ - Mac OS;
- ☐ - Solaris;
- ☐ - Outro:.....;

Que critério é que mais valorizam na escolha de um sistema operativo? *

- ☐ – Preço;
☐ – Recomendações do fornecedor;
☐ – O software veio pré-instalado nos computadores;
☐ – Facilidade do utilização;
☐ – Popularidade;
☐ – Disponibilidade de técnicos para dar suportes;
☐ – Segurança;
☐ – Outro:.....;

Dos softwares listados abaixo costumam pagar pela sua licença? *

	Sim	Utilizo sempre mas não pago licença	Utilizo de vez em quando mas sem pagar a licença	Não utilizo
Microsoft Office Word				
Windows (Sistema Operativo)				
Adobe (Pacote completo)				
3DMax				
Enterprise Architect				
Visual Studio				

Dos softwares abaixo, qual é melhor para sua empresa? *

	1 ° É melhor	2 ° É melhor	Não sei dizer
Windows Vs Linux			
Visual Studio Vs Eclipse			
Photoshop Vs GIMP			
IIS (Windows) Vs Servidor Apache			
MS Office Word Vs Open Office			
Chrome Vs Firefox Mozilla			
MS Visio Vs Dia			
Sublime Text Vs EMACS			
3D Max Vs Blender			

Dos softwares abaixo, quais é que utilizam? *

	Não Conheço	Conheço, mas não utilizamos	Conheço e utilizamos regularmente	Conheço e utilizamos frequentemente
Open Office				
Neo Office				
AbiWord				

Thunderbird				
GIMP				
ImageMagick				
Audacity				
Rhythmbox				
Firefox				
Kompozer				
Scribus				
Inkscape				
Blender				
Drupal				
Moodle				
Linux				
Apache				
OpenProj				
GanttProject				
Emacs				
Dia				
Tellico				
Koffice				
Celtix				
Gnumeric				
Eclipse				
Netbeans				

Dos softwares *Open Source* assinalados acima, há alguns que gostariam de substituir por uma alternativa *Closed Source* ? *

[] – Sim;

[] – Não;

Se sim quais:

.....

.....

.....

.....

Se sim diz o motivo dessa tal escolha:

.....

.....

.....

Quais os principais problemas que costumam ter com os softwares *Open Source*? *

[] – Falta de documentação;

- ☐ - *Crashes*;
- ☐ - Vulneráveis;
- ☐ - Difícil de utilizar;
- ☐ - Falta de suporte técnico;
- ☐ - Dificuldade de manutenção;
- ☐ - Difícil de integrar com outros sistemas;
- ☐ - Outro:.....;

No caso de haver uma falha técnica num software *Open Source* são capazes de resolver o problema? *

- ☐ - Sempre;
- ☐ - Geralmente;
- ☐ - Às vezes;
- ☐ - Nunca;

Enumere as principais vantagens que *Open Source* tem sobre *Closed Source*: *

- ☐ - Preço;
- ☐ - Melhor testado;
- ☐ - Mais personalizável;
- ☐ - Relatório de erro mais completo;
- ☐ - Mais estável;
- ☐ - Mais leve (combina com hardwares de baixo custo);
- ☐ - Melhor qualidade;
- ☐ - Outro:.....;

Enumere as principais desvantagens que *Open Source* tem em relação ao *Closed Source*: *

- ☐ - Pouca documentação;
- ☐ - Difícil de fazer manutenção;
- ☐ - Difícil de utilizar;
- ☐ - Interface não user-friendly
- ☐ - Pouca usabilidade;
- ☐ - Incompatibilidade dos programas;
- ☐ - Poucas garantias;
- ☐ - Mudança de versão rápida;
- ☐ - Baixa qualidade;
- ☐ - Outro:.....;

Como classificam a iniciativa *Open Source*? *

- ☐ - Muito Bom;
- ☐ - Bom;
- ☐ - Normal;
- ☐ - Mau;
- ☐ - Muito Mau;

Fazem parte de qualquer comunidade *Open Source*? *

.....
.....
.....

Acham que *Open Source* é uma ameaça para empresas base tecnológica? *

- ☐ – Sim;
- ☐ – Não;
- ☐ – Talvez;

A.2 Questionário às Universidades

Que instituição pertence? *

- ☐ – Universidade Jean Piaget de Cabo Verde ;
- ☐ – Universidade de Cabo Verde (Uni-CV);
- ☐ – ISCEE;

Qual é a sua função? *

- ☐ – Professor;
- ☐ – Gestor de laboratório;
- ☐ – Administrador;
- ☐ – Reitor;
- ☐ – Outro:.....;

Que sistema operativo utilizam nos laboratórios informáticos? *

- ☐ – Windows;
- ☐ – Linux;
- ☐ – Mac OS;
- ☐ – Solaris;
- ☐ – Outro:.....;

Na escolha dos sistemas operativos que critérios é que mais valorizam? *

- ☐ – Preço;
- ☐ – Recomendações do fornecedor;
- ☐ – O software veio pré-instalado nos computadores;
- ☐ – Facilidade de utilização;
- ☐ – Disponibilidade de técnicos para dar suportes;
- ☐ – Popularidade;
- ☐ – Segurança;

☐ – Outro:.....;

Perante um software *Open Source* e um outro *Closed Source* qual escolhe para instalar nos laboratórios? *

- ☐ – *Open Source*;
- ☐ – *Closed Source*;
- ☐ – Depende;

Se depende, aponta o critério?

- ☐ – Preço;
- ☐ – Recomendações do fornecedor;
- ☐ – Facilidade da utilização;
- ☐ – Popularidade;
- ☐ – Segurança;
- ☐ – Disponibilidade de técnicos para dar suportes;
- ☐ – Disponibilidade da aplicação;
- ☐ – Outro:.....;

Costumam receber pedido por parte dos professores para instalarem um *Software Open Source* em vez de uma alternativa proprietária nos laboratórios? *

- ☐ – Sim;
- ☐ – Não;

Se sim, diga os motivos que o professor apresentou?

- ☐ – Por serem mais didáticos;
- ☐ – Permite mais personalização;
- ☐ – Tem mais funcionalidade que a alternativa proprietária;
- ☐ – Mais rápida;
- ☐ – Preço mais baixo ou grátis;
- ☐ – Outra:.....;

No caso de haver uma falha técnica num software *Open Source* você ou outros membros da instituição são capazes de resolver o problema? *

- ☐ – Sempre;
- ☐ – Geralmente;
- ☐ – Às vezes;
- ☐ – Nunca;

A manutenção geral dos softwares do laboratório é feita pelo pessoal interno ou terceiros? *

- ☐ – Interno;
- ☐ – Terceiro (Externo);

Que tipo de software o pessoal/técnico mais se queixa no momento de fazer manutenção? *

- ☐ – *Open Source*;
- ☐ – *Closed Source*;
- ☐ – Ambos;

Entre os softwares *Open Source* e *Closed Source* qual é que mais necessita de manutenção? *

- ☐ – *Open Source*;
- ☐ – *Closed Source*;
- ☐ – Ambos;

Quais os principais problemas que costumam enfrentar com softwares *Open Source*? *

- ☐ – Falta de documentação;
- ☐ – *Crashes*;
- ☐ – Vulneráveis;
- ☐ – Difícil de utilizar;
- ☐ – Falta de suporte técnico;
- ☐ – Dificuldade de manutenção;
- ☐ – Difícil de integrar com outros sistemas;
- ☐ – Outro:.....;

Enumere as principais vantagens que *Open Source* tem em relação ao *Closed Source*: *

- ☐ – Preço;
- ☐ – Mais bem testado;
- ☐ – Mais personalizável;
- ☐ – Relatório de erro mais completo;
- ☐ – Mais estável;
- ☐ – Mais leve (combina com hardwares de baixo custo)
- ☐ – Melhor qualidade;
- ☐ – Outro:.....;

Enumere as principais desvantagens que *Open Source* tem em relação ao *Closed Source*: *

- ☐ – Pouca documentação;
- ☐ – Difícil de fazer manutenção;
- ☐ – Difícil de utilizar;
- ☐ – Interface não user-friendly ;
- ☐ – Pouca usabilidade;

- [] – Incompatibilidade dos programas;
- [] – Pouca garantia;
- [] – Mudança de versão rápida;
- [] – Baixa qualidade;
- [] – Outra:

Dos softwares listados abaixo, costumam pagar pela licença de alguns deles? *

	Sim	Utilizo sempre mas não pago licença	Utilizo de vez em quando mas sem pagar a licença	Não utilizo
Microsoft Office Word				
Windows (Sistema Operativo)				
Adobe (Pacote completo)				
3DMax				
Enterprise Architect				
Visual Studio				

Se utilizam algum software sem pagar a licença identifica o principal motivo:

- [] – Preço da licença exagerado;
- [] – Instituição não possui recursos financeiros suficientes para pagar pela licença;
- [] – O responsável do laboratório é que sugeriu;
- [] – Sempre que necessitamos dos softwares *craqueados*, é fácil de se encontrar;

Dos softwares abaixo, qual seria melhor para sua instituição? *

	1 ° É melhor	2 ° É melhor	Não sei dizer
Windows Vs Linux			
Visual Studio Vs Eclipse			
Photoshop Vs GIMP			
IIS (Windows) Vs Servidor Apache			
MS Office Word Vs Open Office			
Chrome Vs Firefox Mozilla			
MS Visio Vs Dia			
Sublime Text Vs EMACS			
3D Max Vs Blender			

Dos softwares abaixo, quais é que utilizam? *

	Não Conheço	Conheço, mas não utilizamos	Conheço e utilizamos regularmente	Conheço e utilizamos frequentemente
Open Office				

Neo Office				
AbiWord				
Thunderbird				
GIMP				
ImageMagick				
Audacity				
Rhythmbox				
Firefox				
Kompozer				
Scribus				
Inkscape				
Blender				
Drupal				
Moodle				
Linux				
Apache				
OpenProj				
GanttProject				
Emacs				
Dia				
Tellico				
Koffice				
Celtix				
Gnumeric				
Eclipse				
Netbeans				

Dos softwares *Open Source* indicados acima, há alguns que gostariam de substituir por uma alternativa *Closed Source*? *

[] – Sim;

[] – Não;

Se sim, quais:

.....

Se sim, diz o motivo dessa tal escolha:

.....

Fazem parte de alguma comunidade *Open Source*? *

.....
.....
.....

Como classifica a iniciativa *Open Source*? *

- ☐ – Muito bom;
- ☐ – Bom;
- ☐ – Normal;
- ☐ – Mau;
- ☐ – Muito mau;